

Treball de Fi de Grau
Enginyeria en Tecnologies Industrials

Estudi, control i optimització d'un *LEGO Segway*

MEMÒRIA

Autor: David Llevat Pàmies
Director: Jan Rossell Gratacòs
Convocatòria: Juliol 2015



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

Al llarg d'aquesta memòria es desenvolupa l'estudi per a construir, controlar i optimitzar el comportament d'un *Segway*, o pèndul invertit de dues rodes, mitjançant els components cedits per la companyia de joguines LEGO dins el marc dels projectes LEGO *Education*, concretament els LEGO *Mindstorms EV3*. Aquest tipus de pèndul invertit consta de dues rodes motrius que s'encarreguen de desenvolupar l'acceleració necessària per a intentar mantenir el cos del vehicle en una posició d'equilibri inestable. Aquesta, varia en funció del disseny, la dinàmica del vehicle i del terreny sobre el qual es desplaça.

L'estudi presenta una introducció genèrica del problema del pèndul invertit per tal de determinar els paràmetres dimensionals de més importància i, així, desenvolupar un disseny més estable i eficient. Un cop adquirit el model, es desenvolupa una simulació amb l'ajuda del programari matemàtic Matlab i Simulink per tal de determinar les constants específiques del problema i dissenyar control d'estabilitat del *segway* basat en un controlador LQR i complementat amb un pre-compensador i un observador.

Finalment, s'han desenvolupat, verificat i simulat els models amb èxit però, tot i que el controlador LQR presenta una forma senzilla d'aplicar un control a sistemes *MIMO*, la implementació en temps real sobre el robot no ha funcionat tan bé com s'esperava. Algunes de les causes que poden haver portat al comportament inesperat del sistema real són: errors comesos en alguns paràmetres del sistema degut a la baixa resolució del giroscopi i la falta d'una lectura real de l'angle. Es considera, però, que són problemes que s'haurien pogut solucionar d'haver tingut un marge de temps més ampli.

Sumari

RESUM	1
SUMARI	2
GLOSSARI	5
1. PREFACI	6
1.1. Origen i motivació del projecte.....	6
1.2. Requisits previs.....	7
2. INTRODUCCIÓ	8
2.1. Objectius.....	9
2.2. Abast.....	10
3. CONCEPTES TEÒRICS	11
3.1. Origen del problema	11
3.2. Modelat del sistema.....	12
3.2.1. Hipòtesis.....	12
3.2.2. Espai d'estats	13
3.2.3. Modelat.....	14
3.3. Teoria de control	18
3.3.1. Història [5].....	18
3.3.2. Tria del control.....	19
3.3.3. Teoria LQR [7]	20
3.3.4. Afegir Pre-compensador [6]	22
3.3.5. Observador [9]	23

4. ACOBLAMENT DEL ROBOT	25
4.1. Modificacions i altres elements	26
4.2. Obtenció dels paràmetres.....	29
4.2.1. Pes (mp , mc).....	30
4.2.2. Longitud del pèndol (l).....	30
4.2.3. Motors.....	32
4.2.4. Fregaments	34
4.2.5. Força i PWM	37
5. SIMULACIÓ I IMPLEMENTACIÓ	38
5.1. Construcció del model no-lineal i lineal en Matlab Simulink	38
5.1.1. No-lineal	38
5.1.2. Lineal	39
5.1.3. Comprovació del model.....	40
5.1.4. Hipòtesi de linealitat	41
5.2. Construcció de la planta real.....	42
5.3. Construcció del controlador.....	43
5.3.1. Realimentació i Pre-compensador	43
5.3.2. Observador.....	45
5.4. Calibratge del controlador	48
5.4.1. Temps de mostratge	48
5.4.2. Inicialització	48
5.4.3. Calibratge de Q i R	50
5.4.4. Constant FP	51
6. ESTUDI ECONÒMIC	52
7. CONCLUSIONS	53
AGRAÏMENTS	55

BIBLIOGRAFIA	56
Referències bibliogràfiques	56
Bibliografia complementària	57
ANNEX A. INTRODUCCIÓ AL SISTEMA LEGO <i>TECHNIC</i>	58
ANNEX B. ESPECIFICACIONS DELS COMPONENTS ELECTRÒNICS	60
ANNEX C. CODI MATLAB	68

Glossari

GETI: Grau en Enginyeria en Tecnologies Industrials

LQR: *Linear Quadratic Regulator*

q_{ii} : representen els valors de la diagonal de la matriu Q

x_j : representa els valors del vector d'estats

\ddot{x} = acceleració lineal [m/s^2]

$\ddot{\varphi}$ = acceleració angular de les rodes [rad/s^2]

$\ddot{\theta}$ = acceleració angular [rad/s^2]

$\dot{\theta}$ = velocitat angular [rad/s]

θ = posició angular [rad]

F = força o pertorbació del sistema [N]

F_f = força de fregament (rodolament, de les rodes i dels eixos, i aerodinàmic) [N]

m_p = massa del pèndol [kg]

m_c = massa del carro (rodes i possible afegit a aquestes) [kg]

g = gravetat ($9,81 m/s^2$)

l = longitud del pèndol, des de l'eix de les rodes al centre de masses [m]

R = radi de les rodes [m]

μ_e = coeficient de fregament dinàmic als eixos [$kg \cdot m^2/s$]

I = moment d'inèrcia respecte de l'eix z en el punt situat a l'eix de les rodes

T_n = període natural del pèndol [s]

ω_n = freqüència natural del pèndol [Hz]

ξ = esmorteïment del pèndol

tm = temps de mostra [s]

1. Prefaci

1.1. Origen i motivació del projecte

La memòria presentada a continuació està desenvolupada en el context del Projecte Fi de Grau requerit en l'últim quadrimestre dels estudis de Grau en Enginyeria en Tecnologies Industrial a la Universitat Politècnica de Catalunya. Tot i pertànyer a l'àmbit universitari, aquest projecte ha estat desenvolupat dins el programa Erasmus+ Pràctiques amb el suport de l'empresa *LEGO System* a Billund, Dinamarca. El present projecte s'ha desenvolupat fora de l'àmbit de l'empresa perquè s'ha volgut evitar la confidencialitat.

La idea de desenvolupar aquest projecte va néixer al cap d'un mes d'arribar a l'empresa, quan es va tenir la possibilitat d'accedir al material electrònic. Aquest fet, juntament amb l'interès que van despertar les assignatures de Dinàmica de Sistemes, Control Automàtic i Electrònica, va fer que es decidís buscar un projecte on poder emprar aquests recursos i motivacions.

Donat que el GETI ha estat generalment un aprenentatge teòric, ha estat un gran al·licient poder desenvolupar un projecte amb un final de caire pràctic. Es creu que es una experiència molt positiva aprofitar l'oportunitat per desenvolupar un projecte pràctic, doncs a nivell teòric les coses generalment funcionen, però a l'hora de la implementació real les coses generalment fallen al principi. És per aquest motiu que s'ha volgut aprofitar aquest projecte per agafar autonomia i experiència, doncs en un futur no hi haurà un professor recolzant-nos.

A l'utilitzar el material cedit per l'empresa, la durada del projecte va quedar lligada a l'estada en aquesta. Al final, s'ha comptat amb una mica menys de 4 mesos per desenvolupar el present estudi. Aquest projecte es va considerar adequat a aquestes condicions perquè l'abast és fàcilment modificable: des d'eliminar la implementació fins a afegir funcions extres al robot.

Per últim, afegir que una de les coses més importants que s'ha après cursant aquests estudis ha estat: el desenvolupament d'una visió crítica a l'hora d'enfrontar problemes i l'aplicació dels conceptes bàsics apresos per entendre idees i mètodes no encontrats abans. Per aquesta raó, s'ha triat desenvolupar certes parts del projecte amb mètodes no estudiats anteriorment, tot aprofitant l'ocasió per aprendre nous conceptes i enfocaments per a solucionar problemes.

1.2. Requisits previs

Per tal de programar el robot, s'ha decidit utilitzar llenguatge Matlab i Simulink perquè tenen un paquet de comunicació amb LEGO *Mindstorms EV3* que donarà més versatilitat al càlcul matemàtic que el programa oficial de LEGO. Per tant, seran necessaris coneixements de programació en Matlab i en Simulink.

Per entendre els mètodes desenvolupats són necessaris coneixements bàsics d'àlgebra lineal i de teoria de control.

2. Introducció

El *Segway HT* va ser inventat per Dean Kamen i presentat l'any 2001. El *Segway Human Transporter* és un vehicle individual que funciona de forma intuïtiva balancejant el pes del cos. Ha estat una revolució que ha desenvolupat en interessants reflexions sobre el futur del transport i la mobilitat sostenible [1]: vehicle individual, elèctric, lleuger, ràpid, àgil, bona autonomia; cal realment utilitzar un utilitari per desplaçar-se per la ciutat?

Aquestes reflexions han provocat que des de la presentació del producte, hi hagi hagut un auge en la publicació d'articles i projectes basats en l'optimització del control dinàmic d'aquest producte, que no deixa de ser l'aplicació directa del problema clàssic del pèndol invertit.

Però, en aquest projecte s'utilitzarà un *Segway* construït amb LEGO (evidentment a escala reduïda), per tant, és pertinent presentar el material utilitzat per al desenvolupament del projecte. L'empresa ha cedit un conjunt complet de LEGO *Mindstorms EV3*. Aquest producte va néixer dins els programes de LEGO *Education*, que busquen introduir les joguines a l'aprenentatge des de preescolar fins a l'escola de secundària per a fer-lo més senzill, atractiu i entenedor. Entre molts productes, LEGO fa 17 anys que va començar a desenvolupar un conjunt de productes per a introduir la robòtica als seus productes; anys més tard, ha arribat a les aules d'una forma senzilla i interessant a través de LEGO *Mindstorms*. Aquest producte consta de tres tipus d'instruments a banda dels *bricks Technic*¹, que seran utilitzats al llarg d'aquest estudi² (Figura 1):

- Actuadors: hi ha dos tipus de motors, els quals tenen codificadors angulars integrats per tal de poder fer un seguiment de les rotacions del motor sense

¹ Per a una breu introducció al sistema de construcció LEGO *Technic*, dirigir-se a l'Annex A. Introducció al sistema LEGO *TECHNIC*

² Per a les especificacions dels components dirigir-se a l'Annex B. Especificacions dels components electrònics

necessitat de programar i calibrar-los anteriorment.

- Sensors: es pot trobar un giroscopi d'un sol eix, un sensor d'infrarojos, un sensor d'ultrasons, un sensor de contacte i un sensor de colors/lluminositat.
- *Brick EV3*: última versió del microcontrolador que utilitza el conjunt per tal de controlar el sistema, llegir informació i executar els programes que s'hagin desenvolupat.

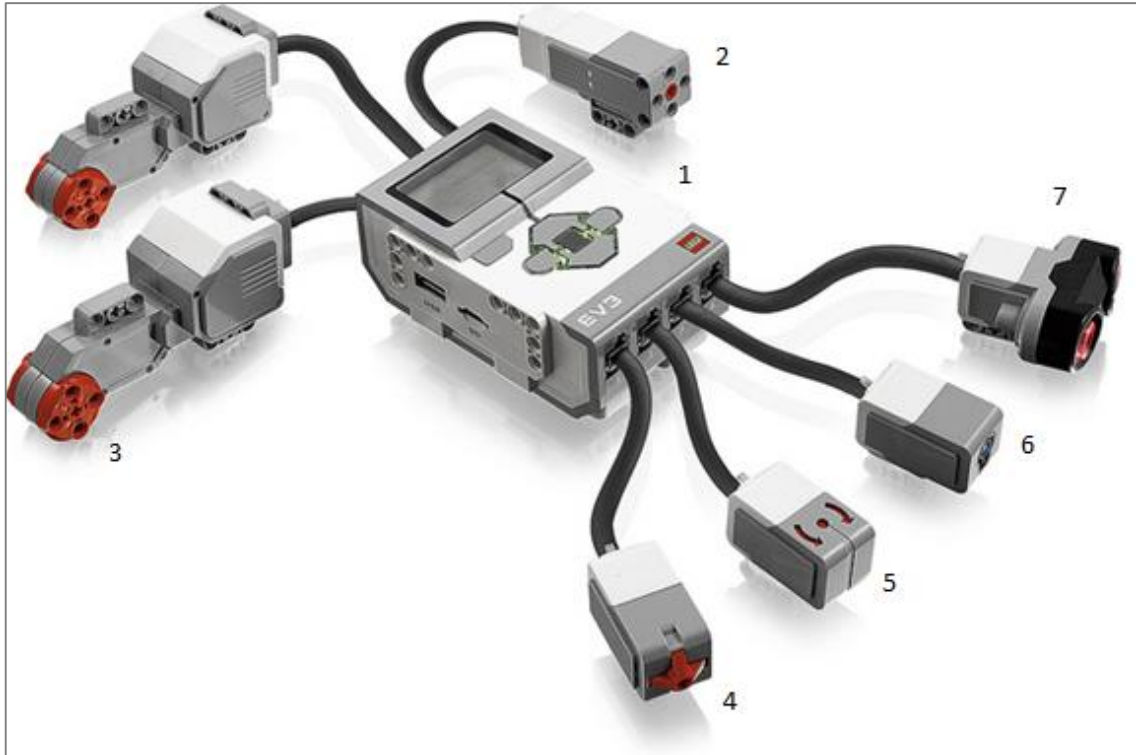


Figura 1. Components electrònics de LEGO EV3. 1- Brick; 2- Motor petit; 3- Motor gran; 4- Sensor contacte; 5- Giroscopi; 6- Sensor color; 7- Sensor infraroig.

2.1. Objectius

Els objectius principals d'aquest projecte són:

- ser capaç de desenvolupar les equacions matemàtiques i de control necessàries per a representar el comportament del sistema,
- ser capaç de simular el comportament del *Segway*,
- ser capaç de trobar els paràmetres reals del sistema i triar correctament les constants de disseny del control per a obtenir els resultats desitjats,

- ser capaç d'implementar un control que sigui capaç de mantenir el robot en posició vertical i a la posició desitjada.

2.2. Abast

Aquest projecte vol desenvolupar l'estudi, simulació i implementació en temps real del control dinàmic d'un *Segway*. Com s'ha explicat a la motivació del projecte, el desenvolupament d'aquest depèn de l'estada a l'empresa i, per tant, l'abast del projecte es podria veure modificat excloent-ne la part de la implementació en temps real. Finalment, s'ha aconseguit fer la implementació en el robot i un estudi de la resposta del sistema real, tot i que ha faltat temps per ajustar els paràmetres adequadament.

3. Conceptes teòrics

3.1. Origen del problema

En aquest punt s'explicarà l'origen del problema que es tracta durant aquest treball. Per això, hem de començar amb la definició d'un terme emprat sovint en aquest treball. Fins al moment, s'ha utilitzat el terme *Segway* sense donar-ne una descripció acurada de en què consisteix. Segons es cita a la pàgina oficial de *Segway* Espanya [2]:

“La tecnología Smart Motion™ de Segway consiste en una red inteligente de sensores, componentes mecánicos y sistemas de control, que permiten al Segway PT mantenerse en equilibrio y desplazarse sobre dos ruedas.

En el momento en que se suba, cinco giroscopios y dos sensores de inclinación detectarán el cambio de terreno y la posición del cuerpo a una velocidad de 100 veces por segundo, más rápido de lo que puede pensar su cerebro”

Tot i ser una descripció prou acurada, per a l'objecte d'estudi és important apuntar que en robòtica s'entén per *Segway* a un sistema carro-pèndol on s'ha eliminat el carro i, per tant, presenta la forma d'un *Segway* amb les mateixes funcions d'aquest, però sense una persona muntada a sobre guiant-lo.

Així doncs, es pot trobar l'origen d'aquest projecte en el problema clàssic del pèndol invertit. La finalitat d'aquest problema és mantenir una massa en equilibri per sobre el punt de rotació, és a dir, en una posició d'equilibri inestable i a la posició desitjada. En el cas del carro-pèndol, el carro és l'encarregat de desplaçar la base del pèndol per mantenir-lo en equilibri, és a dir, en la base triada (Figura 2), l'angle del pèndol nul.

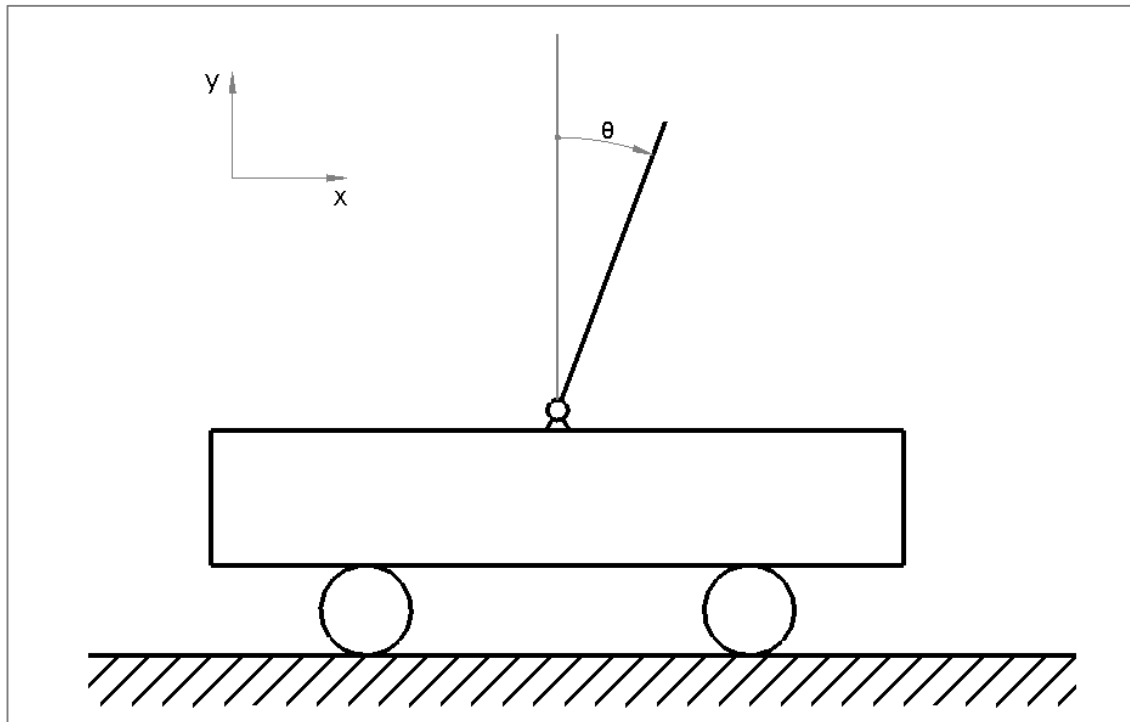


Figura 2. Esquema bàsic del sistema carro-pèndol.

3.2. Modelat del sistema

El problema consisteix en un sistema no-lineal amb dos graus de llibertat (angle del pèndol i posició del carro) força complex sobre el qual s'han desenvolupat diverses solucions al llarg del temps atenent tant als mètodes de representació com a les hipòtesis realitzades. Per una banda, el problema es pot representar mitjançant equacions Lagrangianes o espai d'estats. D'altra banda, es troben tota una sèrie de simplificacions que es poden o no assumir en funció del grau de complexitat que es vulgui afegir al problema, com ara: no considerar la inèrcia de les rodes o els fregaments, la modelització del pèndol com una massa puntual, una barra homogènia o un cos.

3.2.1. Hipòtesis

S'ha decidit triar un conjunt d'hipòtesis que seran utilitzades per a desenvolupar les equacions del moviment (aquestes són analitzades en apartats posteriors un cop es tinguin les dades numèriques):

- Es negligeix la inèrcia de les rodes.

- El parell de reacció roda-motor es considera nul.
- Els fregaments seran considerats fins que siguin analitzats a l'apartat 4.2.4.
- Es modela el pèndol com una barra amb una massa distribuïda homogèniament.
- El sistema és mantindrà a prop dels 0°.
- El sistema és lineal al voltant del punt d'equilibri en un rang de $\pm 20^\circ$.

3.2.2. Espai d'estats

Finalment, s'ha decidit escollir l'espai d'estats com a representació del model. S'ha triat aquesta representació perquè és més adequada a l'hora de treballar amb sistemes MIMO i presenta avantatges per introduir-hi un sistema de control (que s'expliquen posteriorment a l'apartat de teoria); a més a més, l'espai d'estats utilitza una representació de les equacions més familiar amb el que s'ha estudiat durant el grau.

L'espai d'estats consisteix a grans trets [3] en models que empren variables d'estat per descriure un sistema amb un conjunt d'equacions diferencials amb p entrades, q sortides i n variables d'estat; les variables d'estat són magnituds físiques que caracteritzen el sistema a estudiar i formen els eixos de l'espai vectorial de l'espai d'estats. Si les equacions són lineals, invariants en el temps i de dimensions finites, llavors es poden escriure en forma matricial.

Per motius que s'explicaran més endavant, es necessita linealitzar les equacions de moviment, fet que permetrà implementar un sistema matricial que prendrà la següent forma:

$$\dot{x}(t) = A(t) \cdot x(t) + B(t) \cdot u(t) \quad \text{Eq. 1}$$

$$y(t) = C(t) \cdot x(t) + D(t) \cdot u(t) \quad \text{Eq. 2}$$

on,

$x(t)$ és el vector d'estat.

$y(t)$ és el vector de sortides.

$u(t)$ és el vector de control o d'entrades.

$A(t)$ és la matriu dinàmica amb dimensió $n \times n$. Indica com afecta l'estat actual al canvi d'estat.

$B(t)$ és la matriu de control amb dimensió $n \times p$. Determina com l'entrada del sistema afecta al canvi d'estat.

$C(t)$ és la matriu d'observació amb dimensió $q \times n$. Dóna la relació entre l'estat del sistema i l'estat de sortida.

$D(t)$ és la matriu d'acció directa amb dimensió $q \times p$. Permet que l'entrada del sistema modifiqui directament la sortida.

Un cop obtingut el sistema s'hauran de comprovar dues coses: que el sistema és controlable i observable. Aquest pas és necessari per assegurar que no hi hauran problemes d'implementació del sistema de control.

Que un sistema sigui controlable implica que donat un estat inicial i en un temps finit, es pot arribar a un estat final utilitzant les entrades adequades. Per saber si un sistema lineal i invariant en el temps és controlable, és condició necessària que

$$\text{rang}[B \ AB \ A^2B \ \dots \ A^{n-1}B] = n \quad \text{Eq. 3}$$

L'observabilitat d'un sistema implica que donat un estat de sortida, aquest conté prou informació per predir quin ha estat l'estat inicial del sistema. Per saber si un sistema lineal i invariant en el temps és observable, és condició necessària que:

$$\text{rang} \begin{bmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^{n-1} \end{bmatrix} = n \quad \text{Eq. 4}$$

3.2.3. Modelat

Aplicant les hipòtesis esmenades (apartat 3.2.1), el model del *Segway* s'assimila a un sistema carro-pèndol, doncs les rodes tenen una relació amb el pèndol únicament d'articulació i, per tant, es poden considerar com a carro. Per aquest motiu, s'ha decidit utilitzar com a base per a les equacions del moviment l'esquemàtic i plantejament exposat en el document realitzat per Razvan V. Florian [4] del *Center for Cognitive and Neural*

Studies (Coneural, Rumania). S'empra aquest document perquè presenta una reformulació de l'estudi d'Andrew G. Barto, que ha estat un dels de més importància però que presentava alguns errors.

Tot seguit es presenta el procés per aconseguir les equacions del moviment (Figura 3):

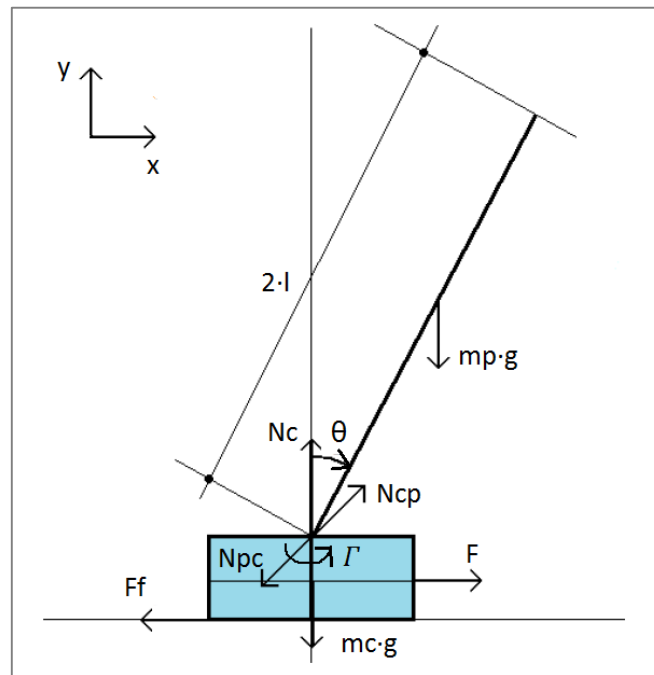


Figura 3. Esquema de forces sobre el sistema carro-pèndol

Aplicant la segona llei de Newton sobre les forces al carro i al pèndol obtenim:

- per al carro,

$$F - F_f - N_x^{pc} = m_c \cdot \ddot{x} \quad \text{Eq. 5}$$

$$N_c - m_c \cdot g - N_y^{pc} = 0 \quad \text{Eq. 6}$$

- per al pèndol,

$$N^{cp} - m_p \cdot g = m_p \cdot a_p \quad \text{Eq. 7}$$

on,

$$a_p = \ddot{x} + \ddot{\theta} \times r_p + \dot{\theta} \times (\dot{\theta} \times r_p); r_p = l \cdot \begin{pmatrix} \sin \theta \\ \cos \theta \\ 0 \end{pmatrix} \quad \text{Eq. 8}$$

Llavors, l'equació per al pèndol i agafant com a referència la mateixa que al carro, queda:

$$\begin{Bmatrix} N_x^{cp} \\ N_y^{cp} \\ 0 \end{Bmatrix} - m_p \cdot g \cdot \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} = m_p \cdot \left(\ddot{x} \cdot \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} + \ddot{\theta} \cdot \begin{Bmatrix} \cos \theta \\ -\sin \theta \\ 0 \end{Bmatrix} - \dot{\theta} \cdot l \cdot \begin{Bmatrix} \sin \theta \\ \cos \theta \\ 0 \end{Bmatrix} \right) \quad \text{Eq. 9}$$

Aplicant la segona llei de Newton una altra vegada, es fa el sumatori de moments sobre el pèndol en el punt de rotació,

$$r_p \times (m_p \cdot g) - \mu_e \cdot \dot{\theta} - r_p \times (\ddot{x} \cdot m_p) = I \cdot \ddot{\theta} \quad \text{Eq. 10}$$

desenvolupant,

$$\left[m_p \cdot g \cdot l \cdot \sin \theta - \mu_e \cdot \dot{\theta} = \left(\frac{4}{3} \cdot m_p \cdot l^2 \right) \cdot \ddot{\theta} + \ddot{x} \cdot l \cdot m_p \cdot \cos \theta \right] \cdot \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} \quad \text{Eq. 11}$$

Substituint l'eix corresponent de l'Eq. 9 en l'Eq. 5 i aïllant, s'obté la primera equació de moviment,

$$\ddot{x} = \frac{F - F_f - m_p \cdot (\ddot{\theta} \cdot l \cdot \cos \theta - \dot{\theta}^2 \cdot l \cdot \sin \theta)}{m_c + m_p} \quad \text{Eq. 12}$$

Substituint l'Eq.12 en l'Eq.11 s'obté la segona equació de moviment,

$$\ddot{\theta} = \frac{(m_c + m_p) \cdot \left(g \cdot \sin \theta - \frac{\mu_e}{m_p \cdot l} \right) - \cos \theta \cdot (F - F_f + m_p \cdot l \cdot \dot{\theta}^2 \cdot \sin \theta)}{l \cdot (m_c + m_p) \cdot \left(\frac{4}{3} - \frac{m_p \cdot \cos^2 \theta}{m_c + m_p} \right)} \quad \text{Eq. 13}$$

Un cop obtingudes les equacions de moviment s'observen 3 no-linealitats: les funcions trigonomètriques del sinus i del cosinus i la velocitat angular que apareix elevada al quadrat. S'ha basat la linealització en la hipòtesi especificada a l'inici d'aquest apartat que diu: el sistema es mantindrà a prop dels 0°. Aquesta hipòtesi implica que es pot utilitzar el desenvolupament de les sèries de Taylor al voltant del 0 en les funcions trigonomètriques. D'altra banda, si el pèndol es manté a prop del punt d'equilibri, el més freqüent és que la velocitat angular sigui petita, per tant, en elevar aquesta al quadrat és pot considerar gairebé nul·la. Així doncs, la linealització queda:

$$\theta \approx 0 \Rightarrow \begin{cases} \cos \theta \approx 1 \\ \sin \theta \approx \theta \\ \dot{\theta}^2 \approx 0 \end{cases}$$

$$\ddot{x} = \frac{F - F_f - m_p \cdot l \cdot \ddot{\theta}}{m_c + m_p} \quad \text{Eq. 14}$$

$$\ddot{\theta} = \frac{(m_c + m_p) \cdot \left(g \cdot \theta - \frac{\mu_e \cdot \dot{\theta}}{m_p \cdot l} \right) - F + F_f}{l \cdot (m_c + m_p) \cdot \left(\frac{4}{3} - \frac{m_p}{m_c + m_p} \right)} \quad \text{Eq. 15}$$

Ara que ja es coneixen les equacions linealitzades ja es pot procedir a construir les matrius de l'espai d'estats. Inicialment s'havia plantejat l'espai d'estats per a un sistema de dues entrades i dues sortides (posició i velocitat angular), doncs eren les úniques lectures directes que es podien obtenir del sistema. Aquest plantejament ha resultat ser força inestable i es creu que ha estat degut a la precisió del giroscopi i a que s'estava intentant controlar un paràmetre amb la seva derivada (l'angle amb la velocitat angular), que té una evolució temporal més ràpida. Finalment s'ha usat un espai d'estats amb 4 entrades i 4 sortides (posició, velocitat, angle i velocitat angular), estimant aquells valors que no es poden conèixer directament. El sistema presenta aquest aspecte:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & 0 & 1 \\ 0 & 0 & c & d \end{bmatrix} \cdot \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ e \\ 0 \\ f \end{bmatrix} \cdot [u] \quad \text{Eq. 16}$$

$$\begin{aligned} \ddot{x} = \theta \cdot a + \dot{\theta} \cdot b + F \cdot e = & \theta \cdot \left(\frac{-m_p \cdot g}{(m_c + m_p) \cdot \left(\frac{4}{3} - \frac{m_p}{m_c + m_p} \right)} \right) + \dot{\theta} \cdot \left(\frac{\mu_e}{l \cdot (m_c + m_p) \cdot \left(\frac{4}{3} - \frac{m_p}{m_c + m_p} \right)} \right) + \\ & + F \cdot \left(\frac{1}{(m_c + m_p)} + \frac{m_p}{(m_c + m_p)^2 \cdot \left(\frac{4}{3} - \frac{m_p}{m_c + m_p} \right)} \right) \end{aligned} \quad \text{Eq. 17}$$

$$\begin{aligned} \ddot{\theta} = \theta \cdot c + \dot{\theta} \cdot d + F \cdot f = & \theta \cdot \left(\frac{g}{l \cdot \left(\frac{4}{3} - \frac{m_p}{m_c + m_p} \right)} \right) + \dot{\theta} \cdot \left(\frac{-\mu_e}{m_p \cdot l^2 \cdot \left(\frac{4}{3} - \frac{m_p}{m_c + m_p} \right)} \right) + \\ & + F \cdot \left(\frac{-1}{l \cdot (m_c + m_p) \cdot \left(\frac{4}{3} - \frac{m_p}{m_c + m_p} \right)} \right) \end{aligned} \quad \text{Eq. 18}$$

La matriu C és una matriu identitat de rang 4 i la matriu D és nul·la.

3.3. Teoria de control

La finalitat d'implementar un control al sistema, és estabilitzar el pèndol en la posició vertical i a la posició desitjada. A fi de triar el mètode ideal, és fa un breu recorregut en els antecedents de la teoria de control i una recerca de mètodes utilitzats en antecedents a aquest projecte.

3.3.1. Història [5]

L'ésser humà té una fixació per trobar la manera de no haver de parar atenció a certs esdeveniments del seu voltant per poder-se dedicar a altres activitats. A l'antiga Grècia ja es troben exemples d'artefactes mecànics que tenien com a finalitat principal repetir un procés al llarg del temps sense la necessitat d'un observador humà. N'és un exemple el despertador de Plató, el qual es basava en omplir un got amb aigua a cabal constant en el qual hi havia un flotador amb boles a sobre; quan aquest s'omplia del tot, les boles queien i funcionaven com alarma. Fent un salt en la història, caldria remarcar el regulador de Watt, el qual va permetre regular la velocitat de les màquines de vapor durant la revolució industrial.

Però aquests exemples només mostren la necessitat d'un sistema que permeti controlar processos, no la existència d'una Teoria de Control Automàtic. De fet, els mètodes de transformacions matemàtiques necessàries s'estaven desenvolupant durant la mateixa època per Laplace i Fourier i, una mica més tard, Cauchy completaria les bases amb els teoremes de la variable complexa. I citant la *Historia de la Ingeniería de Control* de la Universitat de Saragossa [5]:

"El trabajo que presenta Maxwell en 1868: "On Governors" [Thaler 74], puede considerarse como el origen de la Teoría de Control. En él presenta su criterio de estabilidad para sistemas lineales dinámicos e invariantes. En este trabajo Maxwell establece una diferenciación entre Regulators ó Moderators (los conocidos actualmente como reguladores proporcionales) y Governors(reguladores con acción integral)."

A finals dels anys 30 es tenien els primers controladors de 3 termes PID i el que es considerarà l'estructura clàssica. En aquells anys no eren gaire utilitzats en la indústria donat el preu i la falta de comprensió en el seu funcionament. L'espenta que necessitaven va sorgir amb la Segona Guerra Mundial i l'intent de desenvolupar un problema anomenat: *problema del control de tir*, on es pretenia apuntar i disparar automàticament l'arma. Aquest problema va donar pas al plantejament del control en temps discret degut a que les dades proporcionades pels radars eren discontinúes. Durant la guerra, el matemàtic Wiener va treballar en aquest problema i va desenvolupar la *teoria estocàstica clàssica* on introdueix la presència de soroll en els senyals i el concepte de control òptim on busca minimitzar l'error quadràtic com a criteri de qualitat.

A partir de l'any 1955 apareix un nou corrent anomenat teoria de control modern, el qual es basa en la representació dels sistemes utilitzant variables d'estat y està basada en els conceptes d'estabilitat de Liapunov presentat a finals del segle XIX. Un dels noms més importants possiblement és el de R. Kalman perquè va formalitzar el concepte de control òptim i va resoldre el problema del lineal quadràtic gaussià.

3.3.2. Tria del control

Al llarg de la recerca d'antecedents a aquest projecte, s'ha pogut observar que s'utilitzen diversos sistemes de control. Els més freqüents són un control per controlador PID (*Proporcional, Integrador i Derivatiu*), controlador freqüencial i LQR (*Linear Quadratic Regulation*). D'aquests, els dos primers s'han estudiat al llarg del grau i s'està especialment familiaritzat amb el controlador PID.

Aquests tipus de control es poden dividir per una característica principal: el tipus de sistema que tracten. Tant el controlador PID com el controlador freqüencial tracten sistemes SISO (*Single Input Single Output*) i, recordem que un dels objectius d'aquest projecte és controlar tant, l'angle del robot com la posició. Tot i no ser la solució més òptima, és possible implementar aquests tipus de control dues vegades consecutives i sincronitzar-los per aconseguir els propòsits esmenats.

D'altra banda es troben els mètodes de control com l'LQR que tracten sistemes MIMO (*Multiple Input Multiple Output*) i que, per tant, estan dissenyats per a aquesta finalitat. Tot i que aquest mètode no s'ha estudiat al grau, resulta atractiu d'estudiar i com cita *Control Tutorials for Matlab & Simulink* [6]:

*"[...] it is much easier to control multi-input, multi-output systems with the state-space method than with the other methods we have presented."*³

Així doncs, s'estudiarà i implementarà un control LQR al llarg del desenvolupament d'aquest projecte.

3.3.3. Teoria LQR [7]

Aquest tipus de control s'utilitza en sistemes dinàmics lineals i té el propòsit de minimitzar la funció cost quadràtica. Aquest mètode està especialment dirigit a solucionar problemes de control òptim en espais d'estats continus (motiu pel qual s'ha triat aquest tipus de representació); a més a més, no és d'elevada complexitat donat que només cal efectuar operacions d'àlgebra lineal.

La forma de l'espai d'estats ha estat exposada a l'apartat 3.2.2. La condició inicial del sistema és $x(0)$. Assumint que tots els estats són mesurables, es busca un control per realimentació de les variables d'estat (RVE) (Eq. 19)

$$u = -K \cdot x + v \quad \text{Eq. 19}$$

que proporcioni unes bones propietats en llaç tancat. El llaç tancat usant aquest control és de l'aspecte (Eq. 20):

$$\dot{x} = (A - B \cdot K) \cdot x + B \cdot v = A_c \cdot x + B \cdot v \quad \text{Eq. 20}$$

A_c : planta del llaç tancat

$v(t)$: nou vector de control

³ Entre el mètodes presentats es troben un controlador PID i un controlador freqüencial.

Per a dissenyar l'RVE òptim cal definir l'índex d'actuació (IA):

$$J = \frac{1}{2} \int_0^{\infty} x^T \cdot (Q + K^T \cdot R \cdot K) \cdot x \, dt \quad \text{Eq. 21}$$

L'objectiu en el disseny òptim és seleccionar la RVE K que minimitzi l'índex d'actuació J . Aquest pot interpretar-se com una funció d'energia, així que, si J és manté petita, l'energia total del sistema de llaç tancat serà baixa; si l'energia del sistema és baixa implica que tant els estats com l'entrada de control tendiran a ser petits.

Fer èmfasi en que si es minimitza J , aquesta ha de ser finita per definició. Tractant-se d'una integral infinita d' $x(t)$ vol dir que, $x(t)$ tendeix a 0 quan t tendeix a infinit. Aquest fet és garantia de què el sistema en llaç tancat serà estable. També implica que totes les variables a controlar tendiran a 0, per tant, si el valor de control és diferent a 0 tindran un error estacionari.

Els estats del sistema han de ser assolibles per a que l'LQR funcioni, per aquest motiu que les matrius Q i R han de ser definides positives. Els valors de Q i R són triats pels dissenyadors del control en funció dels requeriments de disseny (errors permanents inferiors a y , temps d'establiment no superior a z , etc.); els valors de Q i R actuen de forma relativa entre ells, per tant, l'important és l'escala entre els valors més que els valors en si.

Q és una matriu $n \times n$ semidefinida simètrica i positiva on els elements de la diagonal penalitzen l'error dels estats respectivament.

$$Q = \begin{pmatrix} q_{00} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & q_{nn} \end{pmatrix}$$

on q_{ii} és el pes proporcionat a l'estat quan s'avalua la seva contribució a l'índex d'actuació J . En termes generals, quan més grans són q_{ii} , més penalització s'atribueix a l'estat x_i (és fa més petit) per tal de mantenir l'IA J petit. Aquest fet desenvolupa en que l'error de l'estat x_j acceptat per el sistema és inferior i, per tant, els pols en llaç tancat del sistema A_c (Eq. 20) es troben més a l'esquerra en el pla s ; és a dir, el sistema reacciona més ràpidament a valors de q_{ii} majors. Com Q és diagonal, les penalitzacions s'apliquen a cada variable d'estat

independentment.

R és una matriu $m \times m$ definida positiva que determina el cost energètic del control on els valors representen el pes assignat a cada variable de control. Al contrari que amb Q , seleccionar valors elevats de R vol dir que el vector de control $u(t)$ ha de ser baix per mantenir tal de mantenir l'IAJ petit. Un vector de control baix implica que el sistema aplica menys esforç en el control (forces aplicades inferiors), que generalment desenvolupa en pols del sistema A_c (Eq. 20) més lents i, per tant, errors més grans (estats x_j més grans). Per al cas particular d'aquest projecte, només es té control sobre els motors DC del robot; R és de rang 1.

Per trobar K òptima es suposa que existeix una matriu P constant (Eq. 22):

$$\frac{d}{dt}(x^T \cdot P \cdot x) = -x^T \cdot (Q + K^T \cdot R \cdot K) \cdot x \quad \text{Eq. 22}$$

Substituint a l'Eq. 21, dóna:

$$J = \frac{1}{2} \cdot x^T(0) \cdot P \cdot x(0) \quad \text{Eq. 23}$$

Finalment, K òptima ve donada per l'expressió:

$$K = R^{-1} \cdot B^T \cdot P \quad \text{Eq. 24}$$

I P es troba solucionat l'equació algebraica de Riccati en temps continu (Eq. 25):

$$A^T \cdot P + P \cdot A + Q - P \cdot B \cdot R^{-1} \cdot B^T \cdot P = 0 \quad \text{Eq. 25}$$

3.3.4. Afegir Pre-compensador [6]

El control per LQR és capaç de modificar els transitoris del sistema però no corregeix els errors estacionaris de les variables, que en el cas d'estudi apareix en la variable de la posició del robot (que no necessàriament és sempre 0).

A diferència dels mètodes de control per a sistemes SISO, on es realimenta la sortida per obtenir l'error del sistema, amb un controlador LQR s'està realimentant l'error general del sistema modificat per una matriu K de pesos. Per tant, per eliminar l'error estacionari és necessari modificar l'entrada de referència per ajustar-la a la matriu K utilitzada; aquesta modificació es pot fer afegint un guany \tilde{N} després de la referència desitjada.

El guany \tilde{N} s'ha trobat utilitzant la funció *rscale* de *Control Tutorials for Matlab & Simulink* [8], però pot ser trobada per inspecció directa amb les simulacions fent:

$$\tilde{N} = \frac{\text{valor desitjat}}{\text{valor estacionari de la simulació}} \quad \text{Eq. 26}$$

Cal remarcar que aquest pre-compensador es troba fora de la planta en llaç tancat i que per tant, si hi ha alguna pertorbació no contemplada en la planta seguirà havent-hi error estacionari.

3.3.5. Observador [9]

En general, la resposta aconseguida amb la implementació de l'LQR i el pre-compensador és bona però està basada en la hipòtesi de que es pot obtenir una lectura real de totes les variables, que no té per què ser veritat i, en el cas d'aquest projecte no ho és. Tant la velocitat lineal com l'angle són estimacions.

Assumint que es pot estimar l'estat del sistema (si i només si el sistema és observable (Eq. 4), les dinàmiques de l'estimador d'estat estan descrites per la següent equació:

$$\dot{\hat{x}} = A \cdot \hat{x} + B \cdot u + L \cdot (y - \hat{y}) \quad \text{Eq. 27}$$

L és la matriu $n \times m$ de guany de l'observador.

Aquesta equació és similar a l'Eq.1 del control del llaç tancat amb la diferència que l'últim terme afegeix una correcció a l'estimació de l'estat basat en la diferència entre l'estat actual i l'estimat. Analitzant les dinàmiques de l'error en l'estimador es troba que l'error està descrit per:

$$\dot{e} = (A - L \cdot C) \cdot e \quad \text{Eq. 28}$$

Si la matriu $A-L \cdot C$ té valors propis negatius serà estable i l'error tendirà a 0. Donat que es vol utilitzar l'estimació de l'estat com entrada al controlador, es vol que els estats estimats convergeixin més ràpid que el sistema en llaç tancat; d'altre manera no tindria gaire sentit obtenir valors per predir el següent estat un cop ja no són necessaris perquè ja s'hi ha arribat. Per tant, es vol que els pols de l'observador (pols de la matriu $A-L \cdot C$) siguin de l'ordre d'entre 4 i 10 vegades més grans que el pol més lent del sistema de control.

Per trobar la matriu L es poden utilitzar els mateixos procediments que per trobar la matriu K del control donades les similituds entre l'equació de control (Eq. 20) i la matriu d'estimació. Si es considera que a l'invertir una matriu els valors propis no varien, el problema de l'estimador es pot reformular com:

$$(A - L \cdot C) = (A^T - C^T \cdot L^T) \quad \text{Eq. 29}$$

4. Acoblament del robot

Després de diverses iteracions s'ha decidit optar per triar un model de LEGO existent aplicant-hi algunes modificacions. S'ha pres aquesta decisió perquè al llarg del desenvolupament del projecte s'han trobat diverses adversitats que no pertanyen a l'objectiu d'estudi d'aquest, com ara la falta de rigidesa del model que ha portat a un augment de peces utilitzades i, amb això, un augment de pes.

Finalment, i en base a l'estudi realitzat de les equacions del moviment, s'ha decidit emprar com a model el robot *BALANCER* de LEGO⁴. S'ha optat per escollir aquest model degut a les qualitats que presenta:

- Robustesa del model.
- Senzillesa del model (56 peces incloent sensors, motors i el Brick).
- Estructura esvelta.
- Pes reduït. El model (únicament el pèndol, és a dir, no s'inclouen les rodes) pesa 552 g d'on el 75% del pes correspon a elements necessaris (sensors, motors i *Brick*).

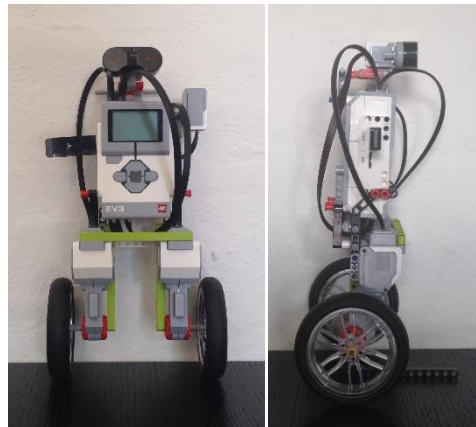


Figura 4. Imatges frontal i lateral del BALANCER.

⁴ Les instruccions per a muntar el robot es poden trobar a: [<http://robotsquare.com/2014/06/23/tutorial-building-balanc3r/>, 5 de juliol de 2015]

4.1. Modificacions i altres elements

A fi de facilitar la realització d'alguns experiments i assegurar-ne la repetibilitat s'ha modificat el model per afegir-hi configuracions. Les configuracions són diferents muntatges que pot adoptar el robot a fi d'acomplir propòsits concrets.

Sobre el cos del robot s'han aplicat dues modificacions que continuen mantenint el model simple i reduït i, alhora, permeten implementar les configuracions de forma ràpida:

- S'ha canviat un dels eixos aspa de 4 mòduls que subjecta el sensor ultrasònic per un de 9 mòduls (Figura 5). Aquesta modificació permet afegir l'Apèndix (descriu a continuació) directament.
- A la part posterior inferior del *brick*, s'ha afegit una "H" (Figura 5). Serveix com a base per al Suport (descriu a continuació).

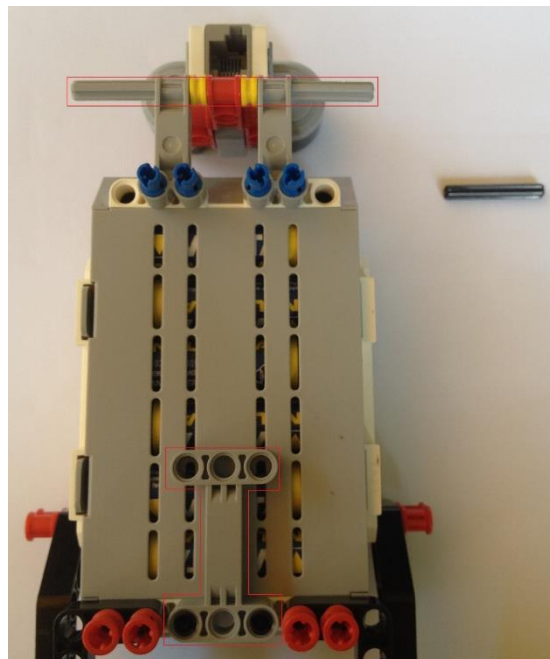


Figura 5. Imatge de les modificacions (ressaltades en vermell) amb l'eix reemplaçat al costat.

Les modificacions que constitueixen les configuracions són elements independents del robot. S'han dissenyat dos elements, l'Apèndix i el Suport:

- Apèndix: consisteix en una roda que gira lliurement subjectada per dues barres angulars tal com es mostra a la Figura 6.



Figura 6. Imatge de l'Apèndix.

- Suport: consisteix en una base que té dos sortints perpendiculars formats per barres en "L". Aquests sortints tenen dues posicions: la primera, A, és amb la "L" cap per amunt i la segona, B, amb aquesta cap per avall (Figura 7). Aquestes posicions permeten tenir un forat femella amb parell o sense respectivament (ressaltat en vermell a la figura esmenada).



Figura 7. Imatge del Suport. A dalt la posició A, a baix la posició B. En vermell es troba ressaltada la geometria del forat.

A partir de les modificacions i elements exposats, es construeixen les configuracions necessàries per als experiments (Figura 8, Figura 10 i Figura 9):

- Configuració 1: es disposarà el robot en posició horitzontal amb l'ajuda de l'Apèndix muntat a la part superior en l'eix aspa de 9 mòduls. Configuració creada per analitzar dinàmiques dels motors lligades al robot sense haver de subjectar-lo per a no interferir en l'experiment. Al llarg de tots els raonaments on s'ha fet ús d'aquesta configuració s'ha tingut en compte de forma conceptual que afegir una roda implica augmentar el fregament de contacte amb el terra.

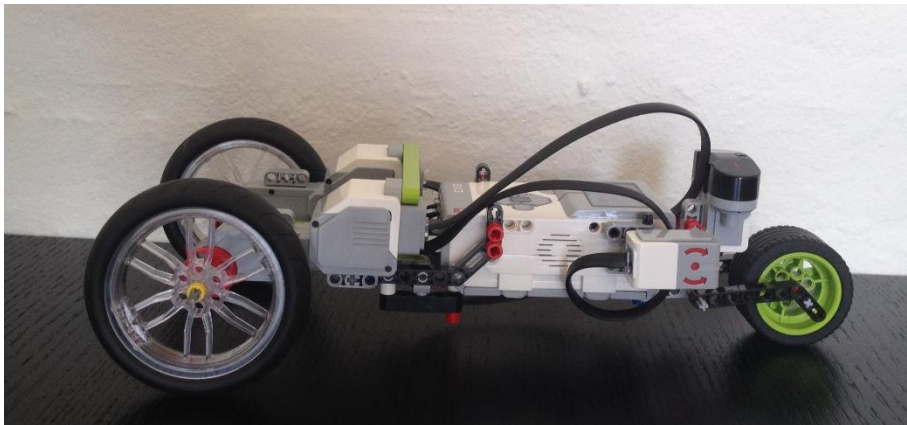


Figura 8. Imatge de la Configuració 1.

- Configuració 2: és igual a la Configuració 1 afegint-hi el Suport (pot ser qualsevol de les dues posicions) en la part inferior enmig de la "H" elevat les rodes motius del terra. L'ús d'aquesta Configuració proporciona una anàlisi pura de les dinàmiques del robot, doncs s'elimina tot tipus de fricció provocat pel moviment del robot.
- Configuració 3: es substitueixen les rodes pel Suport en posició A utilitzant els mateixos eixos. Utilitzant aquesta posició del suport, el robot rotarà al voltant de l'eix dels motors tot activant-los i afegint el fregament dels engranatges interns a la dinàmica.
- Configuració 4: és igual a la Configuració 3 canviant la posició de "L" d'A a B. D'aquesta manera el robot rotarà al voltant de l'eix dels motors sense el fregament produït pels motors.

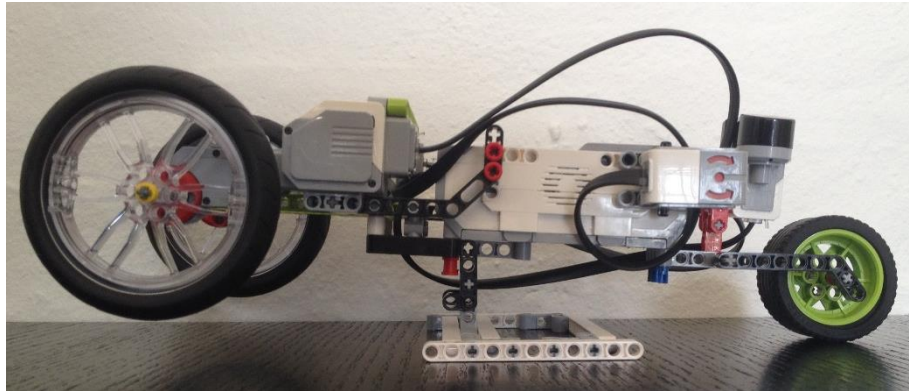


Figura 10. Imatge de la Configuració 2.



Figura 9. Imatge de la Configuració 3 i 4.

4.2. Obtenció dels paràmetres

Donades les equacions de moviment del sistema i el model a utilitzar, s'ha procedit a obtenir tots els paràmetres necessaris per a completar-les i, alhora, corroborar quines suposicions sobre el model es poden prendre. També s'ha experimentat amb els motors amb la finalitat de determinar paràmetres per a la programació posterior.

Degut a la precisió dels sensors emprats, els experiments s'han repetit diverses vegades tot calculant els resultats com a mitjana dels valors obtinguts en aquests; per aquest motiu

els resultats anomenats poden no correspondre amb els gràfics exposats, els quals són una representació (el més acurada possible) dels experiments realitzats.

4.2.1. Pes (m_p , m_c)

Per obtenir el pes total del pèndol (m_p) s'ha procedit a col·locar el robot sense les rodes en una bàscula, obtenint un pes de 552 g.

Com a carro es consideren únicament les rodes del robot, el pes del carro (m_c) és de 80 g.

4.2.2. Longitud del pèndol (l)

A l'hora de realitzar la modelització del sistema per tal de trobar les equacions de moviment s'ha considerat el pèndol com una barra amb una massa uniformement distribuïda. Per tant, seria erroni considerar la longitud eix-centre de masses com la meitat de la barra donat que és només una representació simplificada del pèndol real.

A fi de calcular la longitud del pèndol real, el qual té una forma de capsa amb la massa distribuïda de forma diversa, s'ha utilitzat el següent procediment:

Mitjançant un experiment en el qual el robot ha estat suspès cap per avall utilitzant la Configuració 4 (Figura 11), se l'ha balancejat tot procurant mantenir-lo dins el rang de linealitat (hipòtesi: $\pm 20^\circ$). S'han obtingut les dades de la posició angular per cada instant de temps utilitzant el giroscopi (Gràfic 1), se n'ha extret el període natural (T_n) i substituït en Eq. 30. Finalment el valor de la longitud del pèndol és 16,2 cm.

$$T_n = \frac{1}{\omega_n} = 2\pi \cdot \sqrt{\frac{l}{g}} \quad \text{Eq. 30}$$

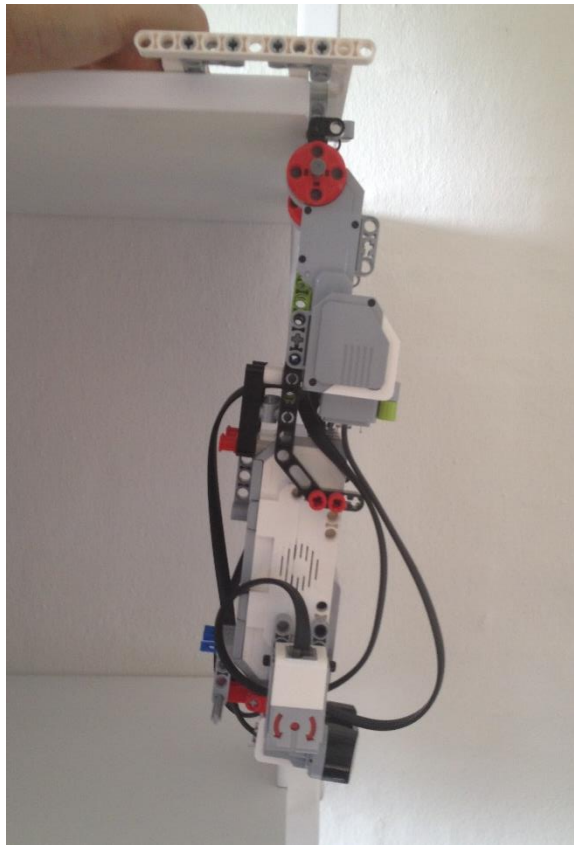
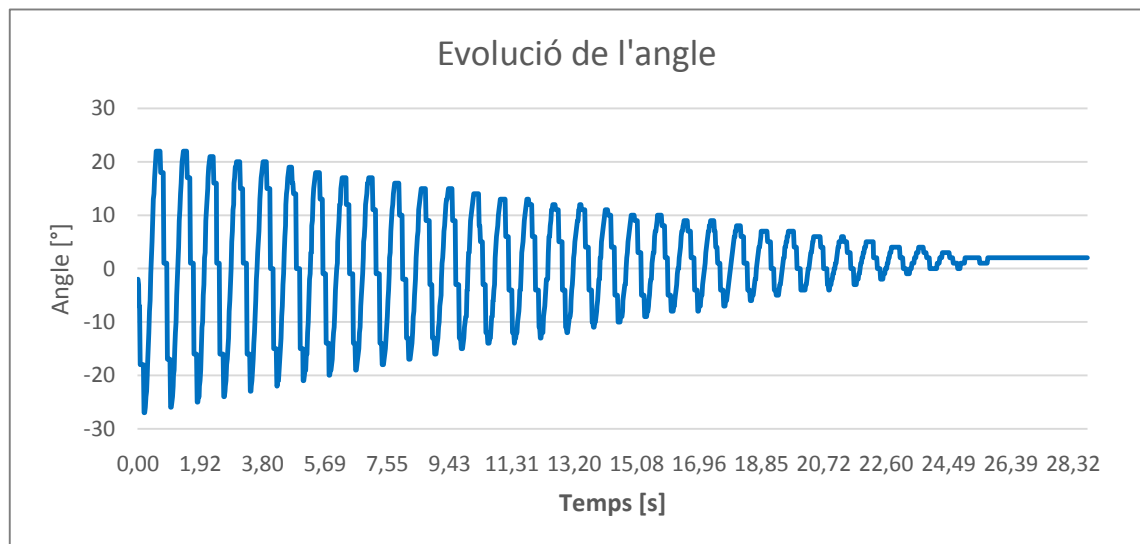


Figura 11. Imatge del posicionament del robot per a trobar la longitud del pèndol.



Gràfic 1. Angle del pèndol (robot) en funció del temps sense fregament als eixos.

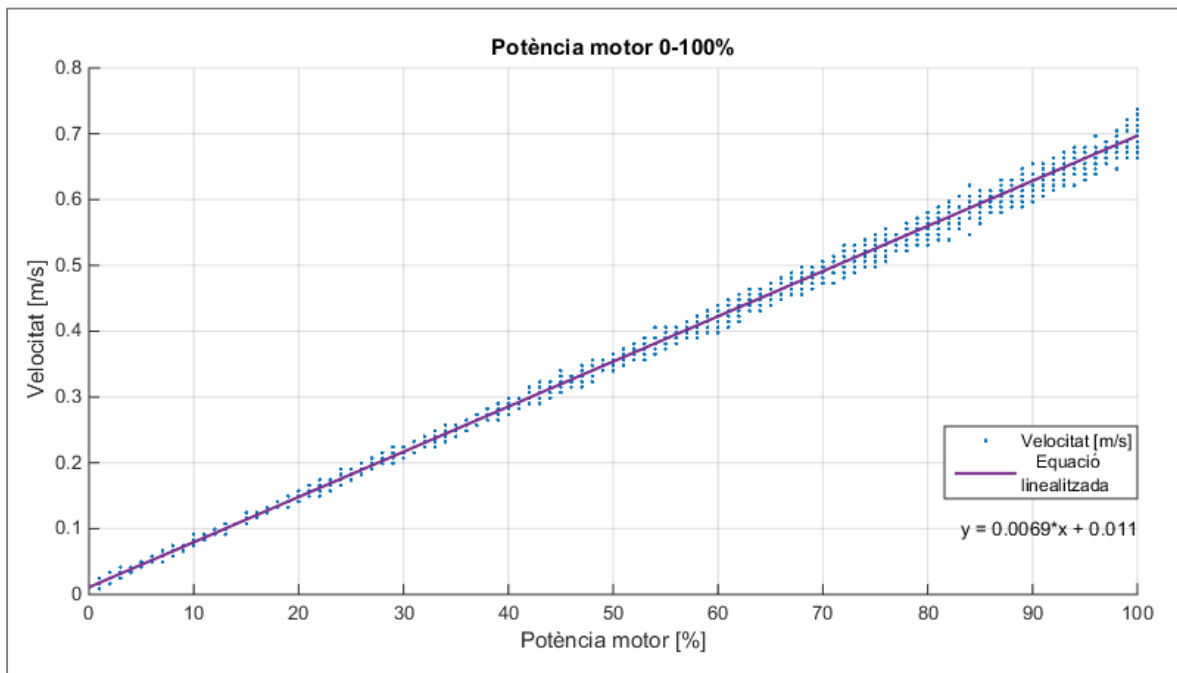
4.2.3. Motors

A fi de determinar la resposta dels motors en relació al robot s'ha procurat fer un estudi extensiu d'aquests amb les eines disponibles.

4.2.3.1. PWM

En una primera instància, exposar que els motors de LEGO MINDSTORMS administren la potència que poden desenvolupar utilitzant una escala que anomenen de “potència” que va des del 0 fins al 100 %, aparentant ser un control PWM (*Pulse Width Modulation*). S'ha estudiat el comportament d'aquesta escala envers la resposta dels motors utilitzant el següent procediment:

- S'ha posicionat el robot en la Configuració 2 (aquest experiment s'ha realitzat a l'aire per qüestions d'espai).
- S'ha incrementat el valor de la potència en un 1% cada 2 segons enregistrant els valors de la velocitat.



Gràfic 2. Variació de la velocitat en funció del % de potència subministrat als motors.

A partir d'aquest assaig, on es pot observar (Gràfic 2) que la relació potència-velocitat és lineal, s'ha corroborat la sospita que el “% de potència” aplicat als motors és un control per PWM.

4.2.3.2. Potència mínima

És important fer èmfasi en que, al haver-se realitzat l'experiment anterior sense contacte amb el terra, s'ha omès el possible fregament de les rodes amb el terra. Per aquest motiu, s'ha realitzat un experiment per determinar quin és el PWM que s'ha d'aplicar al motors (com a mínim) per a què el robot comenci a desplaçar-se. A continuació es determina el procediment seguit:

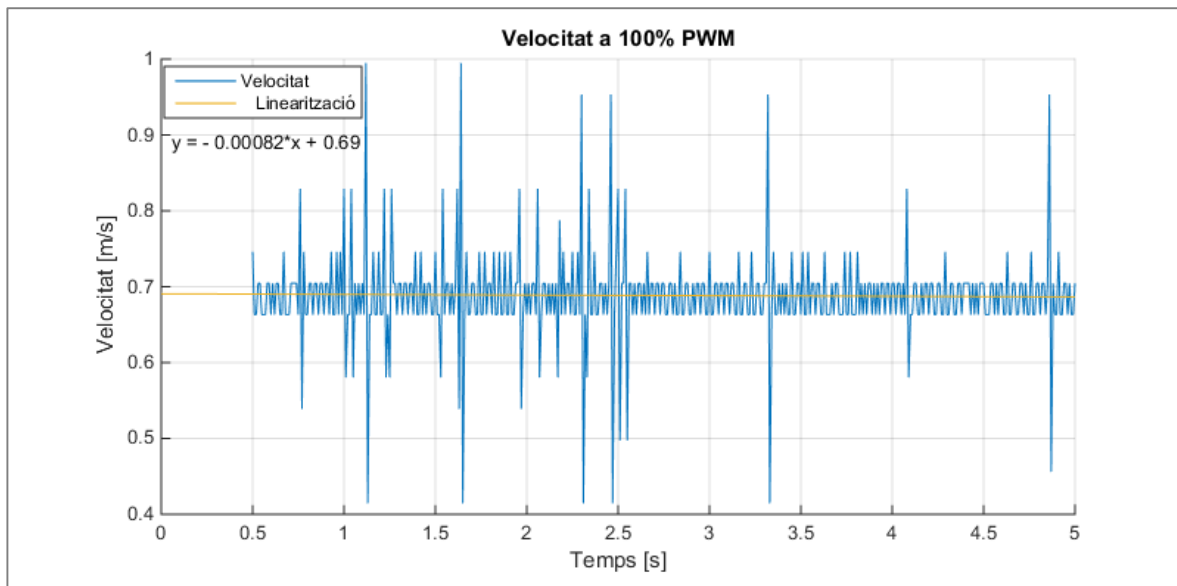
- S'ha posicionat el robot en una superfície plana en la Configuració 1.
- S'ha incrementat el valor de la potència en un 1% i es reinicia el codificador angular dels motors cada 5 segons.
- Un cop el robot llegeix un moviment superior a 90°, s'atura el programa i enregistra l'últim valor de potència emprat.
- La potència mínima necessària és del 3 %.

4.2.3.3. Velocitats i acceleracions

Per tal d'intentar obtenir quin és el rang de velocitats en el que pot treballar el robot (que servirà com a referència i per a futurs càlculs) i quin és l'angle màxim que aquest és capaç de recuperar teòricament, s'han calculat la velocitat màxima i l'acceleració màxima que els motors són capaços de proporcionar arrossegant el robot.

Per a l'obtenció de dades s'ha procedit posicionant el robot en una superfície plana en la Configuració 1. Ambdós motors s'han fixat al 100% de potència des d'un inici i s'ha fet funcionar el robot durant 5 segons tot enregistrant els valors d'angle girat a cada instant de temps (Gràfic 3).

Finalment, s'ha determinat que la velocitat màxima que és capaç d'adquirir el robot és de 0,68 m/s.



Gràfic 3. Velocitat del robot aplicant als motors el 100 % de PWM.

En el cas de l'acceleració, donat que les dades de partida són l'angle girat i els instants de temps, quan s'aplica la derivada per obtenir la velocitat i l'acceleració, els resultats no són del tot acurats i presenten una oscil·lació més pronunciada cada cop que aquesta s'aplica; és per aquest motiu que es considerarà una acceleració mitjana del tram inicial (de 0 s a 0,5 s) d' $1,38 \text{ m/s}^2$ aproximadament.

4.2.4. Fregaments

En un intent de simplificar les equacions de moviment, s'han estudiat els dos fregaments que apareixen en aquestes: el fregament de contacte (F_f) i el fregament als eixos del motor (μ_e).

4.2.4.1. Força de fregament (F_f)

Des del punt de vista de les potències, la potència dels motors és pot considerar que s'inverteix en potència motor (moure el robot) i potència de fregament (vèncer el fregament, majoritàriament el de rodolament).

$$P = P_m + P_f \quad \text{Eq. 31}$$

Donades aquestes consideracions, és pot assumir que la potència mínima necessària per

moure el robot (apartat 4.2.3.2) és la dedicada al fregament, és a dir, el 3% de la potència dels motors.

D'altra banda, en l'experiment portat a terme en apartats anteriors s'han obtingut els resultats de les velocitats màximes (PWM al 100%) en les Configuracions 1 i 2. És pot identificar que la velocitat màxima és més elevada en la Configuració 2 amb un valor de 0,7 m/s (donat que els motors no estan suportant cap càrrega). Per tant, en aquest cas en s'assumeix que $P = P_m$ (s'utilitzarà P_{sf} per referir-se a la potència d'aquest experiment).

Sabent que:

$$P_m = \frac{dW}{dt} = m \cdot \dot{x} \quad \text{Eq. 32}$$

Llavors es troba que:

$$P_f = P_{sf} - P_m \Rightarrow \%P_f = \frac{P_{sf} - P_m}{P_{sf}} \cdot 100 = 2,9\% \quad \text{Eq. 33}$$

Es comprova doncs que en ambdós casos la potència dedicada al fregament és la mateixa (tenint en compte l'exactitud dels processos d'obtenció de dades).

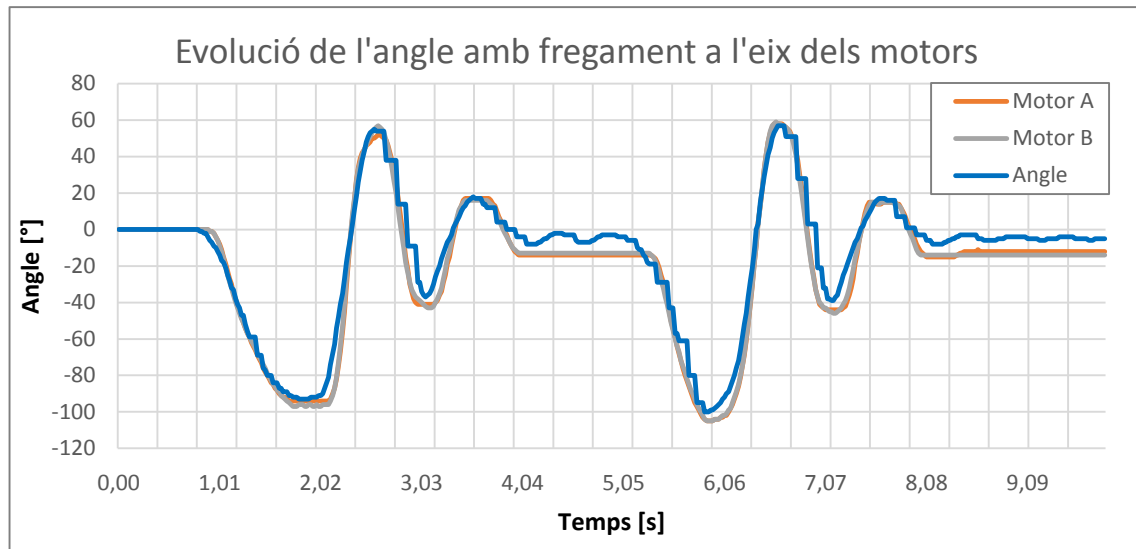
S'ha considerat que sent la Potència de Fregament de l'ordre del 3% de la potència total és prou petita per negligir-la.

4.2.4.2. Constant de fregament als eixos (μ_e)

En un primer instant, al realitzar l'experiment per obtenir la longitud del pèndol es van fixar els eixos (Configuració 3) i es va observar que apareixia un esmorteïment molt pronunciat, és per aquest motiu que s'ha decidit no negligir el fregament als eixos i estudiar-lo.

A fi d'adquirir les dades, s'ha repetit l'experiment exposat en les línies anteriors per a l'obtenció de la longitud del pèndol utilitzant la Configuració 3 (en comptes de la 4); com es pot percebre en el Gràfic 4, tot i elevar el pèndol fins als 80°, aquest s'atura en menys

de cinc oscil·lacions.



Gràfic 4. Angle del pèndol (robot) en funció del temps amb fregament als eixos.

Per obtenir el valor de la constant s'ha procedit a utilitzar l'equació dinàmica del pèndol Eq.34 i transformant-la de forma adient a Eq.35:

$$I \cdot \ddot{\theta} + \mu_e \cdot \dot{\theta} + m_p \cdot g \cdot l \cdot \theta = 0 \quad \text{Eq. 34}$$

Transformada a:

$$\ddot{\theta} + 2 \cdot \omega_n \cdot \xi \cdot \dot{\theta} + \omega_n^2 \cdot \theta = 0 \quad \text{Eq. 35}$$

on,

$$\delta = \ln \left| \frac{x_1}{x_2} \right| \quad \text{Eq. 36}$$

$$\xi = \frac{1}{\sqrt{1 + \left(\frac{2\pi}{\delta} \right)^2}} \quad \text{Eq. 37}$$

$$2 \cdot \omega_n \cdot \xi = \frac{\mu_e}{I} \quad \text{Eq. 38}$$

Com a freqüència natural s'utilitzarà la inversa del període natural obtingut en l'apartat 4.2.2, doncs en l'experiment actual el pèndol es veu afectat pel fregament i, per tant, mostra una realitat distorsionada. Pel que fa a la resta de valors, són tots coneguts excepte la constant de fregament. Resolent s'obté:

$$\mu_e = 0,00769 \text{ kg} \cdot \text{m}^2 / \text{s}$$

4.2.5. Força i PWM

En les equacions obtingudes hi apareix una variable que fa referència a la Força (u), aquesta força (horitzontal) troba el seu punt d'aplicació en l'eix de les rodes i és la responsable de mantenir el robot en equilibri i modificar la posició d'aquest en cas que sigui necessari.

Com es veurà més endavant en la implementació del control, es necessita una relació proporcional (FP) que permeti, donada una força, trobar el PWM que s'ha d'aplicar als motors. En coneixença de que el PWM és una relació lineal, que s'ha d'aplicar un mínim d'un 3% de potència i que la potència dels motors en les especificacions teòriques⁵ és de 14 W s'ha desenvolupat la següent equació:

$$P_{m\grave{a}x} \cdot PWM = \Gamma \cdot \omega_{m\grave{a}x} \cdot PWM = F \cdot r \cdot \omega_{m\grave{a}x} \cdot PWM \Rightarrow F = 4,09 \text{ N} \quad Eq. 39$$

Com l'escala és lineal i anirà del 3 % al 100 %:

$$FP = \frac{PWM}{F} = \frac{100-3}{4,09} \Rightarrow FP = 23,7 \quad Eq. 40$$

⁵ S'ha utilitzat la potencia teòrica donat que no s'ha pogut calcular el parell dels motors i perquè se sap que s'haurà d'ajustar la constant FP en funció de la resposta del robot donades les hipòtesis considerades.

5. Simulació i Implementació

Al llarg d'aquest apartat s'exposarà conjuntament la construcció de la simulació acompanyada de la implementació amb el robot físic per tal de justificar les decisions preses. Les variables i paràmetres dels models han estat implementats en Matlab (es poden trobar a l'Annex C. Codi Matlab); d'altra banda, les simulacions estan implementades amb Simulink perquè com s'ha comentat a la Introducció, es disposa d'un paquet que permet connectar-se a la plataforma de LEGO *Mindstorms*.

5.1. Construcció del model no-lineal i lineal en Matlab Simulink

El primer pas per poder implementar el control sobre el robot físic és la construcció d'un model i la seva posterior comprovació. Anteriorment ja s'han trobat les equacions del moviment, a continuació, s'exposa el procediment que s'ha utilitzat per a construir el model no-lineal i lineal.

5.1.1. No-lineal

Analitzant la finalitat del projecte (crear un control sobre un robot construït amb LEGO *Mindstorms* que sigui capaç d'auto-balancejar-se i de suportar petites pertorbacions) es decideix que la planta ha de tenir com entrada una Força i com a sortida les 4 variables d'estat: Posició, Velocitat, Angle i Velocitat Angular. Així doncs, s'han implementat directament les equacions del moviment (Figura 12):

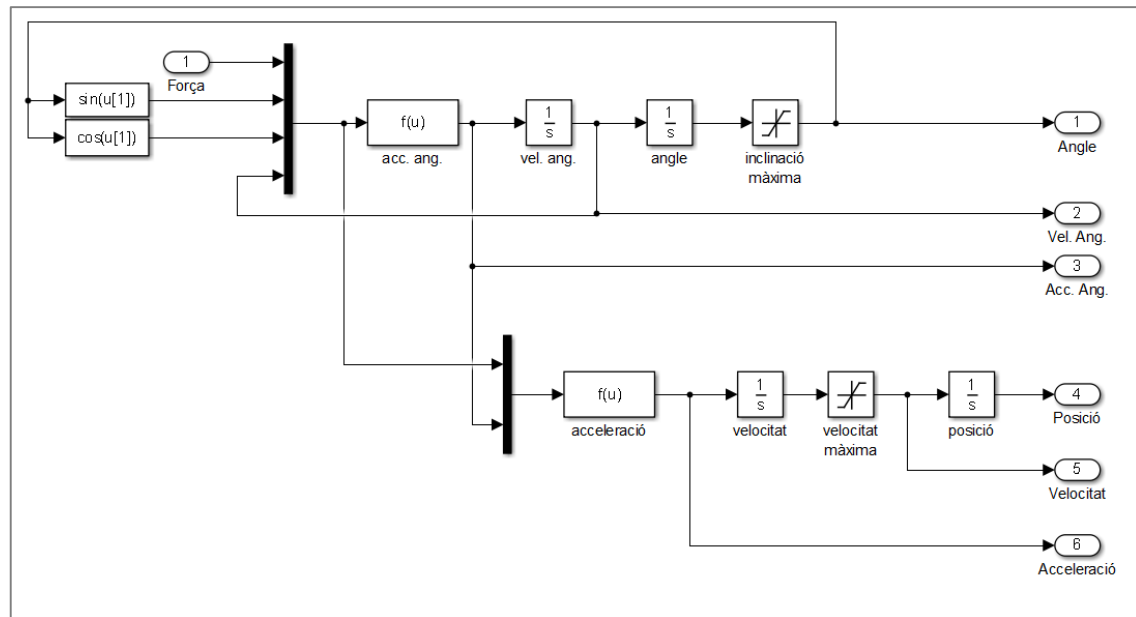


Figura 12. Esquema del sistema no-lineal.

Recordem que les equacions del moviment s'han aïllat en funció de l'acceleració angular i de l'acceleració lineal (Eq. 12 i Eq. 13). Aquestes dues equacions s'han introduït en *Fuction Blocks* i s'han integrat dues vegades per obtenir la velocitat angular i l'angle per una banda, i la velocitat lineal i la posició per l'altre. Comentar que s'han afegit saturadors a l'angle i a la velocitat per tal de limitar el sistema i ser capaç d'observar resultats de forma més acurada.

5.1.2. Lineal

En el cas de l'esquema per al sistema lineal recordem que s'ha d'implementar utilitzant una representació de l'espai d'estats. Les matrius d'aquest espai d'estats s'han desenvolupat a l'apartat 3.2.3, per tant, només cal reconstruir l'equació de l'espai d'estats (Eq. 1):

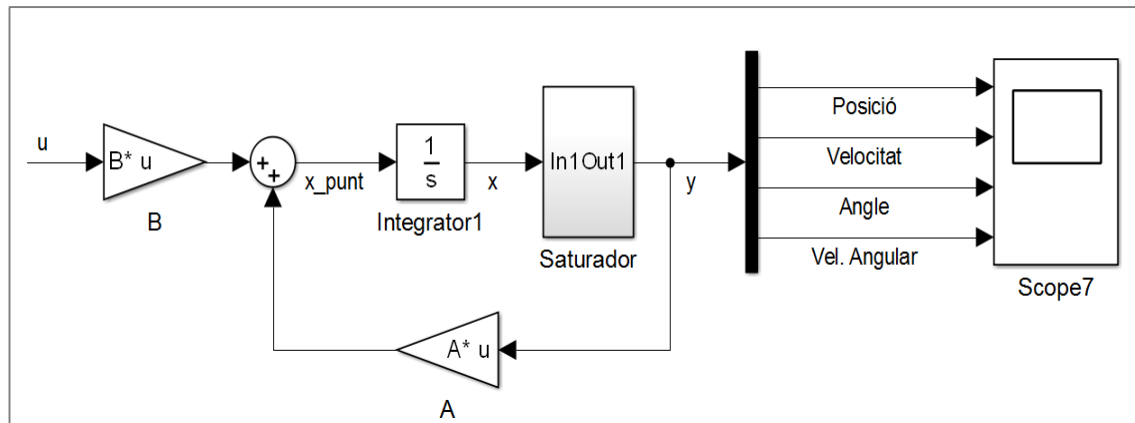


Figura 13. Esquema de l'espai d'estats.

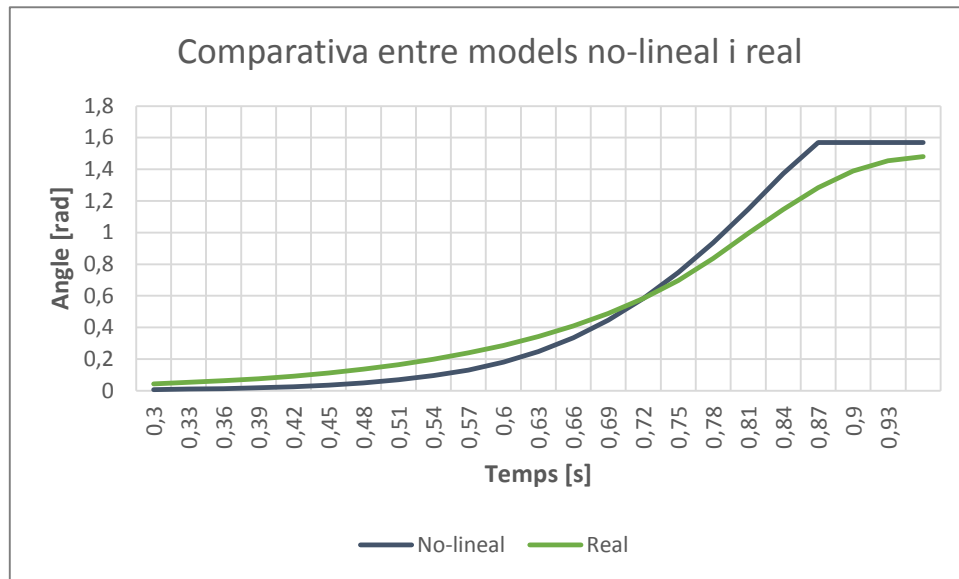
$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -18,77 & 0,16 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 132,63 & -1,16 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 \\ 4,61 \\ 0 \\ -21,39 \end{pmatrix}$$

Assenyalar que la matriu C no apareix a l'esquema perquè és considerada identitat (perquè s'està assumint una lectura dels 4 estats) i llavors tots els estats formen part de la sortida. També s'hi ha afegit un saturador per les mateixes raons que en el cas anterior.

5.1.3. Comprovació del model

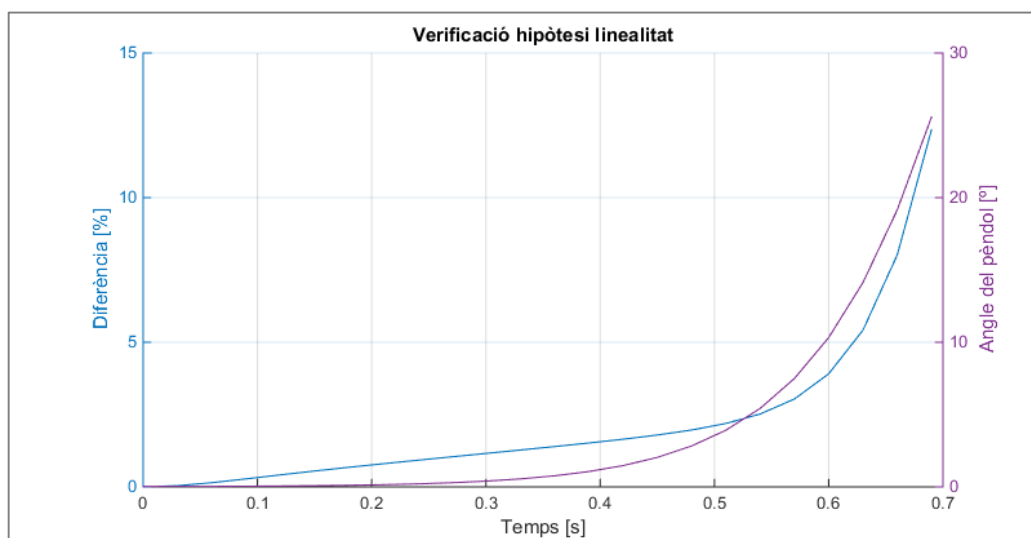
Un cop obtinguts els models de simulació, s'ha comparat el model no-lineal (per ser el més proper a la realitat) amb el prototip real del robot per tal de verificar si és correcte i si les hipòtesis triades fan que aquests difereixin de la realitat. A fi de fer aquesta comprovació, s'ha simulat el model realitzat amb una força de 0 N i s'ha modificat la condició inicial de l'angle a 1° (modificant la condició inicial del segon integrador de la velocitat angular). En el cas del robot, s'ha fet mostrar l'angle actual del robot en la pantalla i s'ha deixat caure quan aquest mostrava la mesura d'1° respecte la vertical. Com es pot observar en el Gràfic 5, el model no-lineal i el real presenten la mateixa tendència.



Gràfic 5. Comparativa entre els models no-lineal i el real.

5.1.4. Hipòtesi de linealitat

Un cop verificat el model no-lineal, s'ha procedit a analitzar la hipòtesi de linealitat en un rang de $\pm 20^\circ$. Per a tal fi, s'han simulat en paral·lel els dos models en les mateixes condicions de l'apartat anterior: 0 N i 1° . El gràfic que es mostra a continuació mostra el % de diferència en base al model no-lineal per a la lectura de l'angle, doncs és el que es distancia més ràpidament i sobre el qual s'ha basat la hipòtesi de linealitat:



Gràfic 6. Comprovació de la hipòtesi de linealitat.

Es pot observar que per a valors al voltant de 20° per al sistema no-lineal, la diferència entre ambdós models és de l'ordre del 8 %. Però per a angles de 10° la diferència cau fins al 3 %. Per tant, es considerarà la hipòtesi com a vàlida.

5.2. Construcció de la planta real

A fi de poder realitzar la implementació del control en el robot, s'ha de transformar la planta teòrica en la planta real. Aquesta planta ha de ser un bloc que transformi la variable de control $u(t)$ en una potència als motors i ha de tenir com a sortida una lectura dels 4 estats. L'aspecte del bloc dissenyat és el següent:

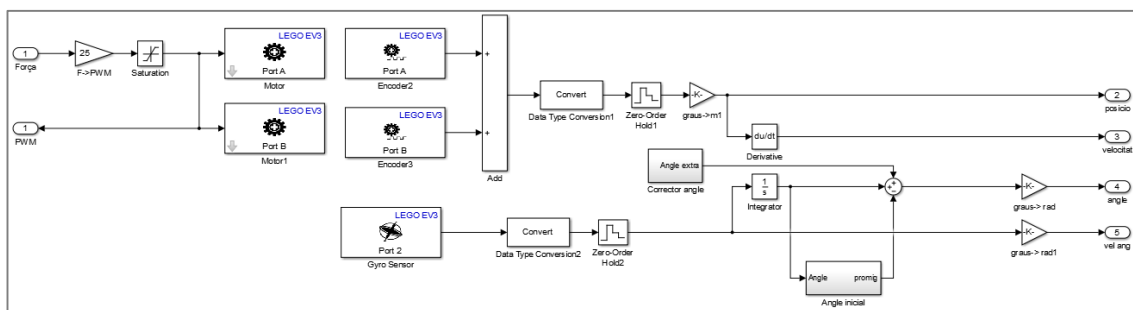


Figura 14. Esquema de la planta real.

El bloc té com entrada la força ($u(t)$) i el primer que es fa és transformar la Força en PWM com s'ha desenvolupat anteriorment (apartat 4.2.5), el valor teòric d'aquesta transformació és $FP = 23,7$. Aquest valor s'ha de saturar a ± 100 , doncs són els valors màxims que accepta el bloc Motor de LEGO *Mindstorms*.

Com ja s'ha deixat entreveure anteriorment, el paquet de Simulink per a LEGO *Mindstorms* no permet utilitzar els sensors en tota la seva magnitud. De fet, només permet la lectura de la rotació del codificador angular dels motors, la velocitat angular del giroscopi i la senyal del comandament a distància. Per tant, s'han estimat els valors de la velocitat i l'angle, derivant i integrant respectivament els senyals llegits. Abans però, se'ls ha aplicat un filtre d'ordre 0 per a convertir les senyals digitals a analògiques, doncs la descripció del sistema s'ha fet així. Tots els senyals s'obtenen en graus i es converteixen a radians perquè és la unitat que utilitza el sistema.

Per la banda dels motors, comentar que tot i rebre la mateixa senyal de potència, la sortida difereix en petita mesura i per aquest motiu es fa una mitjana de les lectures. També dir que el bloc de lectura *Encoder* permet reiniciar la lectura a cada mostra (i també amb senyal extern), que multiplicat per una constant de la forma $(1/\text{temps de mostratge})$ dóna com a resultat la velocitat. S'han comparat la velocitat obtinguda per aquest mètode i derivant la lectura de l'angle; s'ha observat que són pràcticament iguals, per tant, i perquè el programa només permet posar un bloc *Encoder* per motor, s'ha triat llegir l'angle dels motors i derivar-la per obtenir lectura de la velocitat.

Pel que fa als blocs *Angle Inicial* i *Corrector angle* formen part del calibratge i estan explicats a l'apartat 5.4.2.

5.3. Construcció del controlador

Un cop obtingudes la planta lineal i la planta real s'ha procedit a construir el controlador pas a pas. Primer és muntarà el llaç tancat i pels motius que s'explicaran, s'implementarà un observador. Per comprovar que les implementacions han estat realitzades correctament s'ha emprat la planta lineal en un principi i després s'ha canviat per la planta real. Totes aquelles constants que no es comentin al llarg d'aquest apartat, tindran com a valor el predeterminat o teòric desenvolupat al llarg del projecte.

5.3.1. Realimentació i Pre-compensador

Per a construir el llaç tancat només cal agafar la planta i realimentar-la negativament amb una constant de guany K . Aquesta constant s'ha definit a l'apartat 3.3.3 i està dissenyada per a sistemes dinàmics MIMO. El pre-compensador s'ha explicat a continuació de la constant de guany i ens eliminarà l'error permanent sempre que no hi hagin pertorbacions internes.

Considerant les matrius Q i R amb valors iguals a 1, els resultats de les variables d'interès a l'executar el codi són:

$$K = [-1,0000 \quad -1,7576 \quad -16,4470 \quad -1,9221]$$

$$\tilde{N} = -1$$

L'esquema que presenta la simulació és de l'estil (Figura 15):

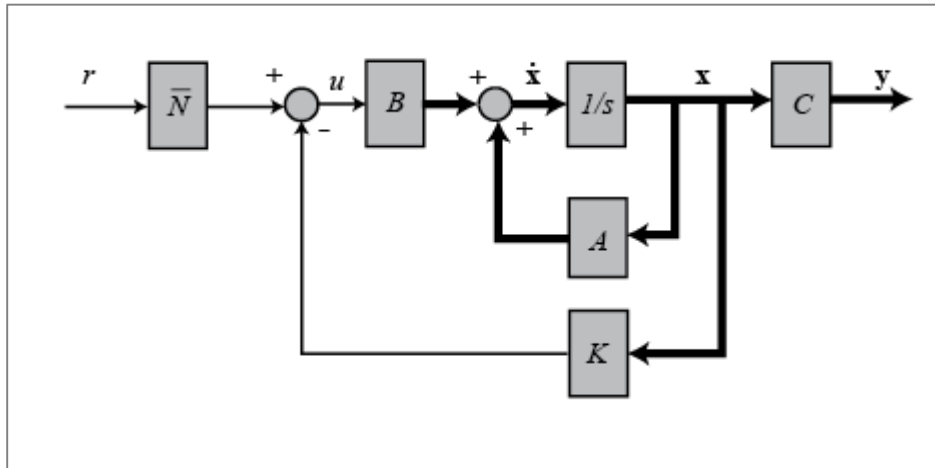
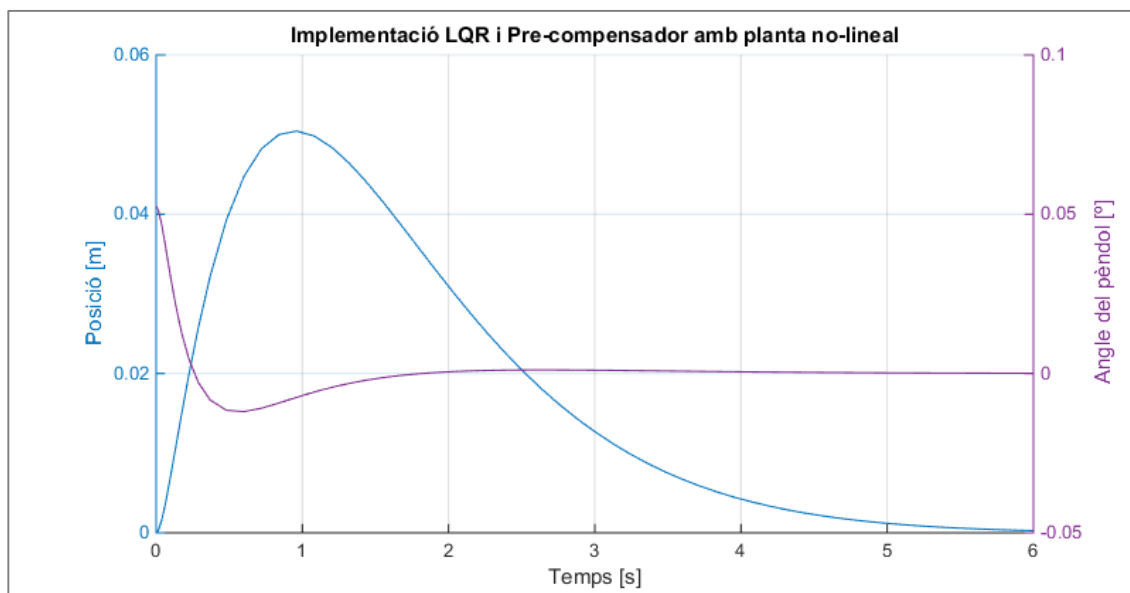
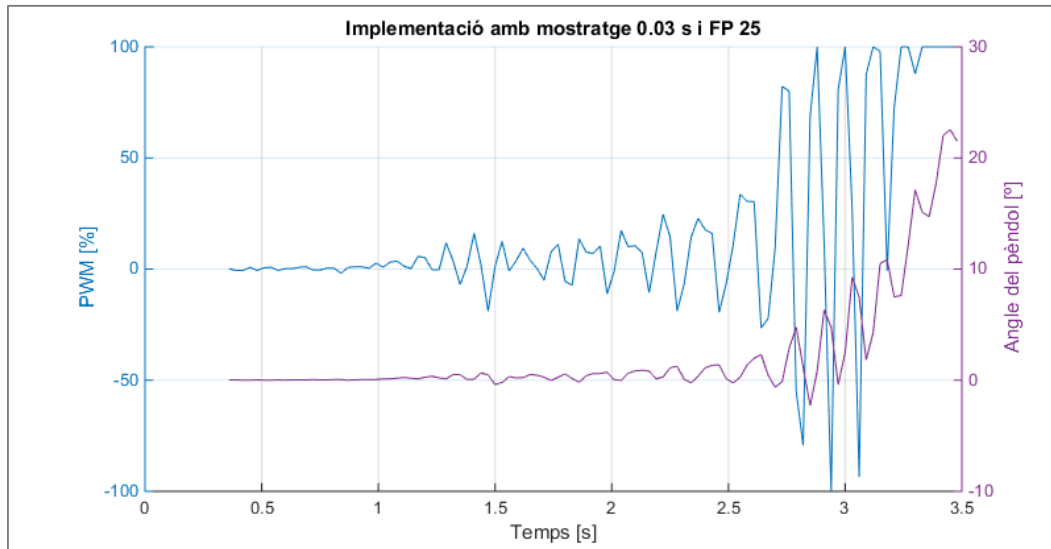


Figura 15. Esquema de la planta amb el controlador LQR i el pre-compensador.



Gràfic 7. Resposta del sistema no-lineal amb la implementació del control LQR.



Gràfic 8. Resposta del sistema real amb la implementació del controlador LQR

Si executem la simulació amb la planta no-lineal (3° com a condició inicial) observem (Gràfic 7) que el pèndol s'estabilitza però tarda 6 segons a assolir la posició desitjada.

Si executem la simulació amb la planta real observem (Gràfic 8) que presenta una tendència a voler mantenir-se en equilibri però cada vegada les oscil·lacions són més pronunciades. Se suposa que el creixement d'aquestes oscil·lacions és produït perquè la realimentació només té en compte l'error actual i, no què passarà un cop s'hagi aplicat la força.

5.3.2. Observador

En base a la suposició que el sistema presenta oscil·lacions creixents que acaben en la pèrdua de l'equilibri perquè el controlador LQR només té en compte l'error actual, s'ha decidit emprar un observador. Com s'ha exposat a l'apartat 3.3.5, la funció de l'observador és estimar el valor del pròxim estat del sistema, és a dir, predir on i com es trobarà el robot en un futur pròxim (passada 1 unitat de *temps de mostratge*) donada la dinàmica actual.

L'esquema de l'observador implementat és:

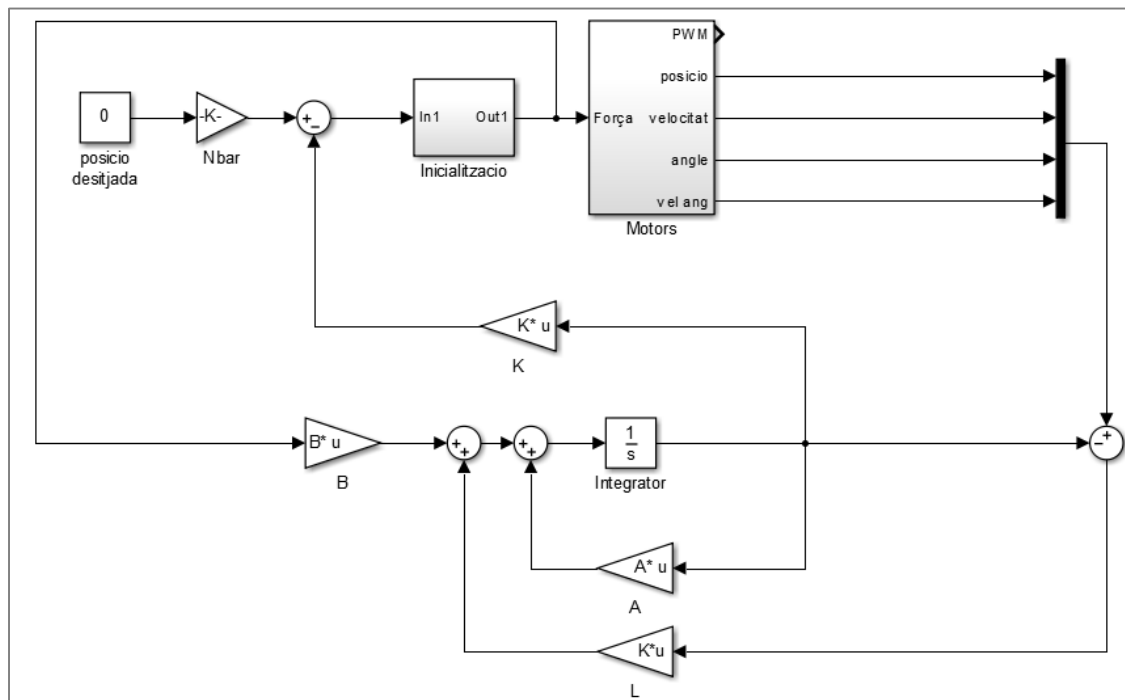


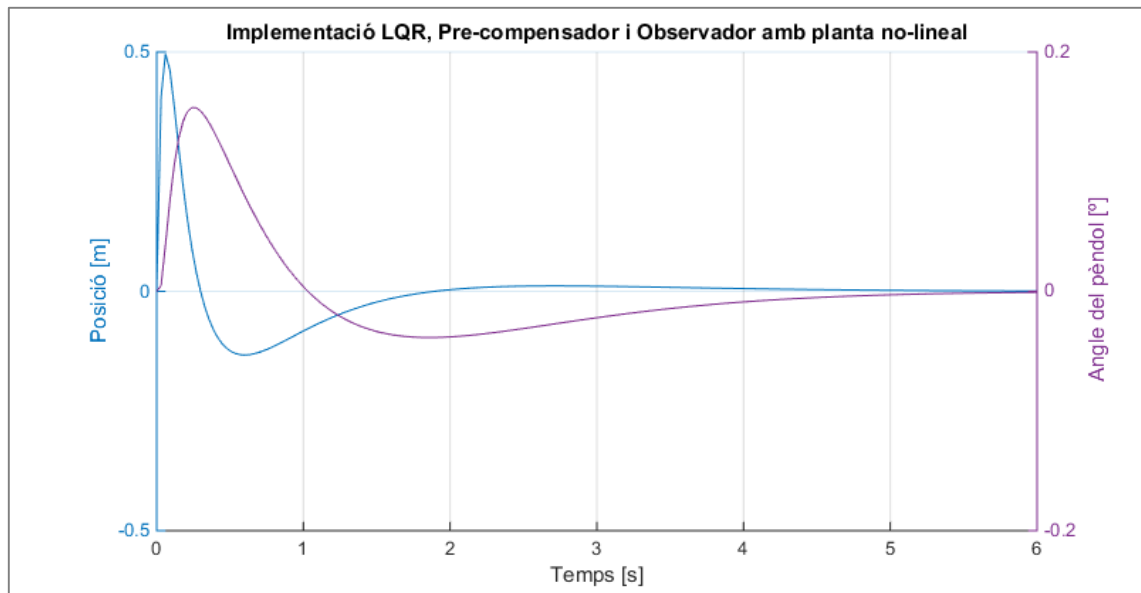
Figura 16. Esquema de la planta amb el controlador LQR, el pre-compensador i l'observador.

Executant el codi que es troba a l'Annex C. Codi Matlab i considerant que totes les constants es troben en els seus valors predeterminat, obtenim el valor de L considerant que els pols d'aquesta tenen una convergència 9 vegades més ràpida que el llaç tancat.

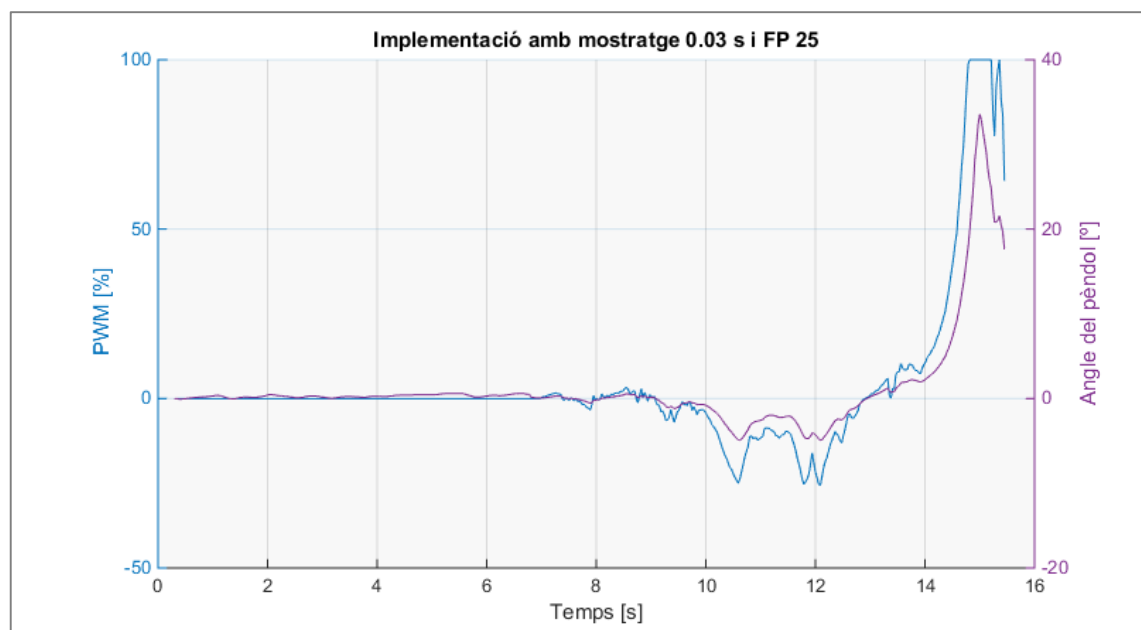
$$L = \begin{pmatrix} 11 & 1 & 0 & 0 \\ 0 & 12 & -18,77 & 0,16 \\ 0 & 0 & 13 & 1 \\ 0 & 0 & 132,63 & 12,84 \end{pmatrix}$$

Si simulem el sistema amb la planta no-lineal, podem observar (Gràfic 10) que tot i mantenir l'LQR estàndard, als 3 segons la posició ja es troba gairebé al punt desitjat. Se'n pot extreure que l'observador agilitza la resposta del sistema.

Si simulem el sistema amb la planta real, podem observar que en comparació al gràfic on apareixien les oscil·lacions creixents (Gràfic 8), la resposta amb l'observador implementat presenta una resposta més suavitzada (Gràfic 9). Amb la resposta obtinguda es pot assumir que la suposició presentada és correcta



Gràfic 10. Resposta del sistema no-lineal amb la implementació del controlador LQR i l'Observador



Gràfic 9. Resposta del sistema real amb la implementació del controlador LQR i l'observador.

5.4. Calibratge del controlador

Al llarg d'aquest apartat s'explicarà el procés de calibratge que s'ha desenvolupat, per tant, s'han simulat totes les proves en el robot físicament.

5.4.1. Temps de mostratge

En un inici s'havia usat un *temps de mostratge* (tm) de 0,01 s, però la resposta presentava un petit retard en l'aplicació de la força, doncs s'apreciava que es començava a aplicar el control quan l'angle del robot era massa elevat. Aquest fet va fer plantejar-se si s'estava utilitzant un tm massa petit. Els sensors poden treballar com a màxim a una freqüència de 1 KHz, per tant, pot ser que el problema vingués donat perquè el *Brick* no té prou potència de càlcul i necessiti un tm més llarg. A fi de comprovar-ho, s'ha implementat el programa amb un tm de 0,001 s, que és el límit dels sensors i s'ha pogut observar que el robot s'ha saturat i portava un retard de més d'1 s. Finalment, s'ha optat per utilitzar un tm de 0,03 s, prou ràpid per reaccionar i prou lent per a que el *Brick* tingui temps de fer els càlculs pertinents.

5.4.2. Inicialització

El bloc de lectura *Gyro Sensor*, no permet reiniciar-lo externament. Per aquesta raó, quan s'inicia el programa, el robot agafa com a angle 0° el punt on es troba en aquell instant (condició inicial de la integral). Per intentar solucionar aquest inconvenient s'ha afegit una fase de inicialització (Figura 17).

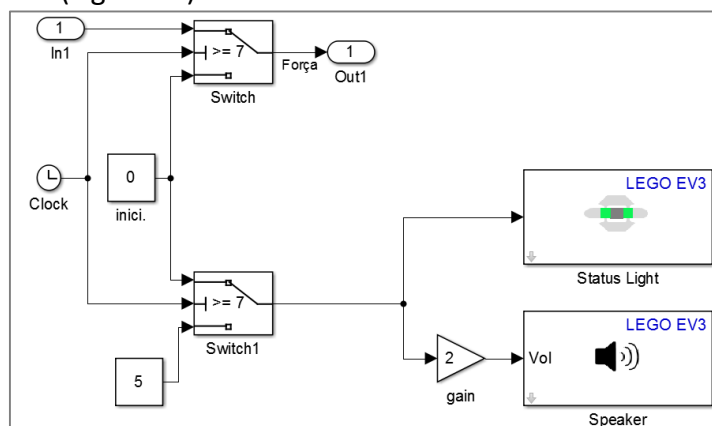


Figura 17. Bloc d'Inicialització.

La fase d'inicialització dura 7 segons i té dues accions: posar el robot en espera i corregir l'angle de referència. Per fer més fàcil saber quan comença i acaba aquesta fase d'inicialització, al llarg dels 7 s sona un to i el *brick* mostra una llum polsant vermella.

Mentre dura la fase, el bloc *Inicialització* (es pot trobar a la Figura 16) substitueix la Força que calcula el control per 0, de manera que els motors estan en espera (Figura 17). Això permet subjectar el robot en la posició d'equilibri per a que el bloc *Angle Inicial* faci una mitjana dels angles llegits (es pot trobar a la Figura 14); aquesta mitjana es converteix a constant quan la fase s'acaba i se suma com a constant correctora a la lectura de l'angle (Figura 19). En cas que no sigui suficient i s'intueixi que el robot té tendència a voler equilibrar-se en una posició no vertical (donat a la inicialització de l'angle), s'ha afegit un bloc anomenat *Corrector Angle* (es pot trobar a la Figura 14) que permet modificar l'angle d'equilibri amb el comandament a distància (Figura 18).

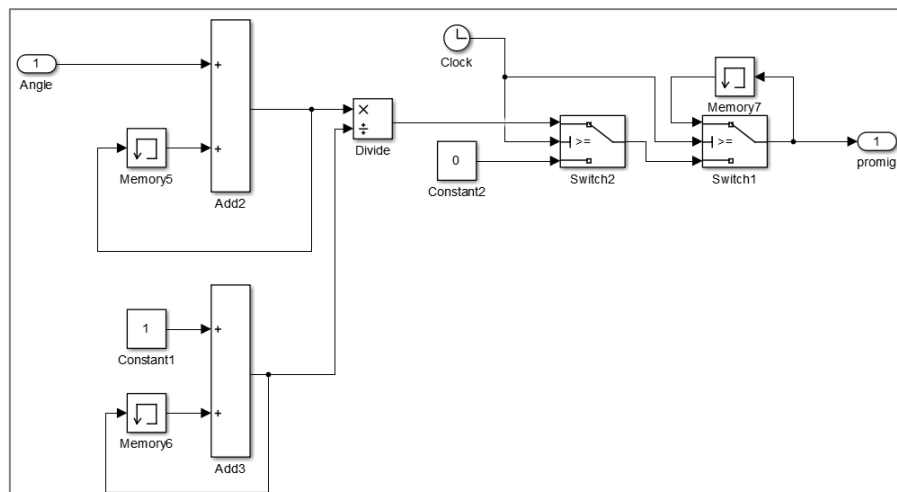


Figura 19. Bloc *Angle inicial*.

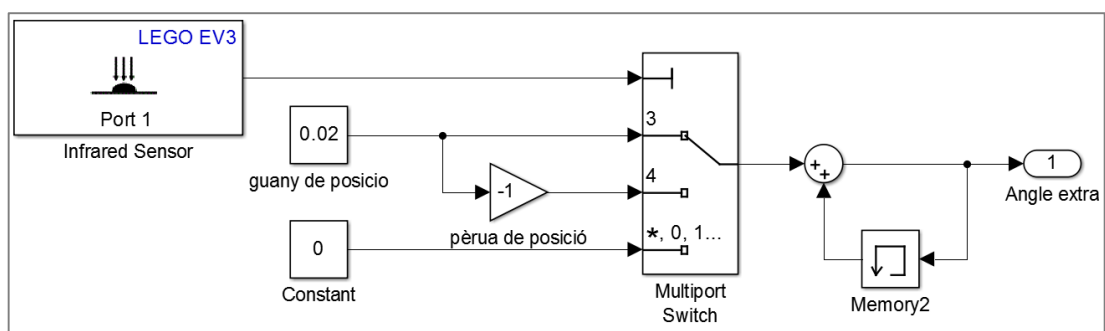
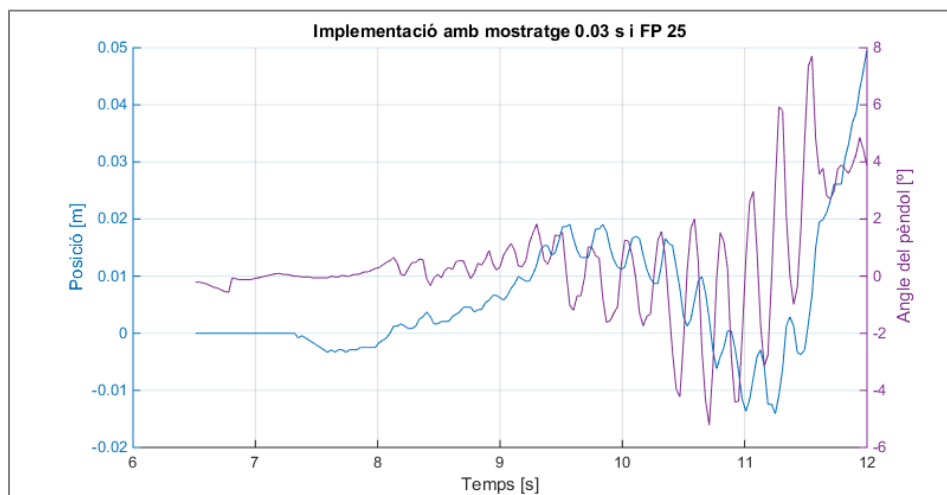


Figura 18. Bloc *Corrector Angle*.

5.4.3. Calibratge de Q i R

Veient que en les execucions prèvies el robot té certs problemes amb l'angle d'equilibri i que quan està al voltant de $10\text{-}15^\circ$ no és capaç de recuperar-se, s'ha decidit incrementar la velocitat de resposta del sistema. Com s'ha explicat als conceptes teòrics, per fer el sistema més ràpid s'han d'augmentar els valors de Q . Augmentar tots els valors de Q per igual no té cap sentit, ja que aquesta ens dóna la oportunitat de controlar el marge d'error que som capaços d'acceptar en cada variable d'estat. En aquest cas, el principal treball del controlador ha de ser per mantenir el robot en equilibri i per això, s'ha augmentat el valor de Q corresponent a la variable d'estat angle, que es troba a $Q(3,3)$.

Secundàriament, s'ha de controlar la posició del robot respecte el punt de partida. El valor de Q per a la posició del robot s'ha trobat que ha de ser força més petit que el de l'angle, doncs pren importància ràpidament dins la dinàmica del control. El valor corresponent a la variable d'estat posició es troba en $Q(1,1)$.



Gràfic 11. Resposta del sistema real amb $R = 1$, $Q(1,1) = 50$ i $Q(3,3) = 1500$.

Fins al moment, els millors resultats han estat obtinguts amb $Q(1,1) = 50$ i $Q(3,3) = 1500$ (Gràfic 11). En aquest cas, la velocitat de reacció del robot és prou bona. Però finalment no s'ha aconseguit calibrar el robot del tot correctament, doncs quan aconsegueix establir-se i necessita tornar a la posició de referència, fa un canvi de direcció massa brusc i intenta tornar massa ràpid. S'ha provat a reduir el valor de $Q(1,1)$, però aquest té un efecte sobre

l'estabilitat del robot i no permet grans modificacions.

5.4.4. Constant FP

El valor teòric de la constant que relaciona la força calculada pel controlador amb el PWM dels motors ha estat fixat en 23,7. Aquest valor però, està basat en els valors teòrics dels motors i a l'hora executar el programa aquest es mou en un rang prou ampli en funció del voltatge de les bateries (des de FP 25 fins a FP 55). No s'ha pogut calcular una relació entre la constant FP i la tensió de les bateries perquè no s'ha disposat de les eines per mesurar aquesta última. Però s'ha descobert que si FP és massa elevat, el control transmet un excés de força provocant una oscil·lació creixent que acaba en la pèrdua de l'equilibri; en cas que FP sigui petit, el robot simplement no té força per equilibrar-se.

6. Estudi econòmic

L'impacte econòmic que ha suposat aquest projecte ha estat nul gràcies als recursos cedits per l'empres i per la universitat. Però , si s'hagués hagut d'adquirir tot el material i eines utilitzades al llarg de l'estudi, aquest hauria resultat ser altament inviable.

El conjunt de *LEGO Mindstorms EV3* té un cost de 350 €.

Però el que realment dispara el pressupost són les llicències de *Matlab & Simulink*, doncs la més bàsica ronda els 6.000 €.

El paquet de suport de *Simulink* per a *LEGO Mindstorms EV3* ha estat gratuïta perquè s'ha utilitzat la versió de prova.

De cara a estalviar-se problemes i tenir cura del medi ambient, comentar que és aconsellable adquirir la bateria recarregable (amb un preu aproximat de 60 €) de què disposa el robot, doncs al utilitzar-lo connectat al wi-fi tot el temps, les piles es consumeixen molt ràpid. En concret s'han utilitzat, 60 piles d'1,5 V en un període de 3 mesos.

7. Conclusions

En aquest apartat s'analitzaran els aspectes del treball que s'han acomplert i també aquells en que no s'han obtingut els resultats esperats, juntament amb possibles causes i solucions.

Ha estat possible accomplir l'estudi de la dinàmica del model i determinar-ne tots els paràmetres. Emprant aquest estudi, s'ha construït un model que ha pogut ser verificat comparant-lo amb el robot real sense aplicar cap control i actuant com a pèndol en caiguda lliure. La verificació dels models (no-lineal i lineal) indica que les hipòtesis plantejades poden ser assumides facilitant els càlculs posteriors sense entrar en una zona de risc.

S'ha aconseguit construir una simulació amb controlador LQR, un pre-compensador i un observador. A nivell teòric, no s'aprecia un canvi important amb la implementació de l'observador, però els efectes s'han fet presents en la implementació en temps real sobre el robot.

Les matrius de control de l'LQR, Q i R , presenten una forma senzilla i clara de com manipular els efectes del controlador sobre la planta. S'ha aconseguit establir el robot fora de la posició desitjada; però no ha estat possible aconseguir que aquest torni a la posició desitjada a una velocitat baixa per a no perdre l'estabilitat.

El fet de necessitar utilitzar el wi-fi per comunicar-se amb el *Brick* provoca un desgast molt ràpid de les bateries. Al no tenir-se una relació entre FP i la tensió de les bateries, el calibratge del robot s'ha complicat.

Les raons per les quals el projecte finalment no ha funcionat adequadament poden ser moltes, però les principals són:

- Tots els paràmetres han estat aconseguits experimentalment, implicant que hi poden haver desviacions en els càlculs que derivin en errors importants. És possible que la baixa resolució dels sensors hagi afectat als paràmetres com I , I o μ_e .

- La baixa resolució del giroscopi (1°) resulta un problema quan la principal necessitat és tenir una lectura acurada. A més a més, s'ha vist que a nivell teòric el giroscopi pot fer lectures de velocitat angular de fins a $440^\circ/\text{s}$, però a velocitats angular altes, aquest perd precisió i conseqüentment, el robot perd referència de l'angle.
- S'ha trobat un valor de Q que acomplia part de les necessitats, però no era efectiu completament. Segurament, de no haver disposat d'un temps definit pel marc del projecte (com s'ha explicat), s'hauria pogut trobar els valors de Q adequats.
- El *temps de mostratge* pot ser que sigui massa elevat i que ajudi a la falta d'estabilitat.

A continuació, s'exposen possible solucions que es poden implementar per tal de millorar el comportament i que no s'han pogut testar pel marc de temps en el qual s'ha desenvolupat el projecte:

- Es pot mirar d'afegir un filtre Kalman al control per a reduir el soroll i aconseguir un valor mig per a les lectures dels estats.
- S'ha intentat implementar una mitjana per blocs de dades amb Simulink, és a dir, tenir com a sortida de les lectures una mitjana cada x mostres pressos. Així es reduiria el soroll i les oscil·lacions en les lectures. Però no s'ha trobat la manera de realitzar aquesta funció.
- Ampliar la quantitat de blocs i/o funcions del paquet de comunicació entre LEGO *Mindstorms EV3* i Matlab Simulink, per ser capaç de llegir angles, voltatge de les bateries, detectar obstacles amb sensor *IR*, etc.

Tot i no haver complert tots els objectius, el desenvolupament d'aquest projecte ha estat una experiència positiva que ha ajudat a consolidar coneixements i a adquirir gran quantitat de nous coneixements en els àmbits del control aplicat a la robòtica.

Agraïments

Es vol fer menció a l'empresa *LEGO System*, per haver donat el suport en tot el que ha estat necessari per a desenvolupar aquest projecte. En especial, al departament de robòtica, *LEGO Mindstorms* per cedir el material.

També anomenar al professor Jan Rossell, al qual se li vol agrair el guiatge efectuat al llarg del projecte.

Bibliografia

Referències bibliogràfiques

- [1] G. ARTÉS, DAVID. *10 años de Segway: reflexiones sobre diseño industrial y movilidad sostenible*. Diariomotor [30 de juny de 2011]. Web.
[<http://www.diariomotor.com/tecmovia/2011/06/30/10-anos-de-segway-reflexiones-sobre-diseno-industrial-y-movilidad-sostenible/>], 29 de juny de 2015]
- [2] SEGWAY ESPAÑA. *Una mirada de cerca*. Web. [<http://www.segway.es/mirada.asp>], 1 de juliol de 2015]
- [3] WHITWORTH, ANDREW. *Control Systems/State-Space Equations*. Wikibooks [10 de febrer de 2014]. Web. [https://en.wikibooks.org/wiki/Control_Systems/State-Space_Equations#Multi-Input.2C_Multi-Output], 15 de març de 2015]
- [4] V. FLORIAN, RAZVAN. *Correct equations for the dynamics of the cart-pole system*. Coneural [10 de febrer de 2007]. Web.
[http://www.coneural.org/florian/papers/05_cart_pole.pdf], 20 de març de 2015]
- [5] UNIVERSIDAD DE ZARAGOZA. ÁREA DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA. *Historia de la Ingeniería de Control*. Web. [<http://automata.cps.unizar.es/>], 2 de juliol de 2015]
- [6] CONTROL TUTORIALS FOR MATLAB & SIMULINK. *Inverted Pendulum: State-Space Methods for Controller Design*. Web.
[<http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum§ion=ControlStateSpace>], 28 de juny de 2015]
- [7] UNIVERSITY OF TEXAS. *Linear Quadratic Regulator (LQR) State Feedback Design*. [31 d'octubre de 2008]. Web. [<http://www.uta.edu/utari/acs/Lectures/lqr.pdf>], 20 de març de 2015]

- [8] CONTROL TUTORIALS FOR MATLAB & SIMULINK. *Function rscale: Finding the Scale Factor to Eliminate Steady-State Error*. [6 de desembre de 1998]. Web. [http://ctms.engin.umich.edu/CTMS/index.php?aux=Extras_rscale , 20 de juny de 2015]
- [9] GEORGE MASON UNIVERSITY. ELECTRICAL & COMPUTER ENGINEERING. *Full-State Observer Notes and Example*. Web. [http://ece.gmu.edu/~gbeale/ece_521/xmpl-521-observer-01.pdf , 26 de març de 2015]

Bibliografia complementària

Com a bibliografia que no està referenciada al llarg de treball però que s'ha utilitzat per a revisar alguns conceptes bàsics s'ha usat els apunts de teoria de les assignatures de Dinàmica de Sistemes i de Control Automàtic.

Annex A. Introducció al sistema LEGO *TECHNIC*

La marca LEGO es dedica exclusivament a produir joguines de construcció. Dins, d'aquestes joguines es troben majoritàriament 2 sistemes de muntatge: el tradicional o de *bricks* i el *TECHNIC*. Però gairebé totes les peces estan dins el sistema general de mesura: mòduls. Un mòdul és la longitud del *brick* 1x1 (Figura A. 1), que equival aproximadament a 8 mm.



Figura A. 1. *Brick* 1x1x3/4

El sistema *TECHNIC*, s'utilitza bàsicament per totes aquelles construccions que necessiten robustesa i/o tenen parts tècniques com ara engranatges o motors. El conjunt de peces bàsiques de que es compon aquest sistema són:

- *Beams* o bigues: són les barres llargues amb nombre imparell de forats, el nombre de forats és el nombre de mòduls de la biga. Els forats de les bigues poden tenir dues geometries diferents:
 - Forat rodó: permet la rotació de tots els eixos en la direcció axial.
 - Forat en creu: només accepta eixos aspa i no permet cap rotació.



Figura A. 2. Biga en forma de L amb els dos tipus de forat.

- Speg o eix: té la funció d'unir dues o tres peces. Té unes petites protuberàncies als extrems per a que els eixos no se surtin dels forats tot sols.



Figura A. 3. Eix simple.

- *Cross-axle* o eix aspa: només proporciona subjecció si s'utilitza un forat creu. Pot agafar més de tres peces si la longitud és l'adequada. Van des de 2 mòduls fins a 15 mòduls.



Figura A. 4. Mostres d'eix aspa.

Totes les peces tenen multituds de variants, però en essència tenen cadascuna alguna de les qualitats descrites. Pot donar-se el cas que algunes peces siguin una barreja de dos dels tipus.

Annex B. Especificacions dels components electrònics

Totes les especificacions presentades a continuació pertanyen al manual de l'usuari de *LEGO Mindstorms EV3*.

EV3 TECHNOLOGY

EV3 Brick


The **Brick Status Light** that surrounds the Brick Buttons tells you the current status of the EV3 Brick. It can be green, orange, or red and can pulse. Brick Status Light codes are the following:

- + Red = Startup, Updating, Shutdown
- + Red pulsing = Busy
- + Orange = Alert, Ready
- + Orange pulsing = Alert, Running
- + Green = Ready
- + Green pulsing = Running Program


You can also program the Brick Status Light to show different colors and to pulse when different conditions are met (learn more about using the **Brick Status Light Block** in the EV3 Software Help).

TECHNICAL SPECIFICATIONS FOR THE EV3 BRICK


- + Operating System—LINUX
- + 300 MHz ARM9 controller
- + Flash Memory—16 MB
- + RAM—64 MB
- + Brick Screen Resolution—178x128/Black & White
- + USB 2.0 Communication to Host PC—Up to 480 Mbit/sec
- + USB 1.1 Host communication—Up to 12 Mbit/sec
- + Micro SD card—Supports SDHC, Version 2.0, Max 32 GB
- + Motor and Sensor Ports
- + Connectors—RJ12
- + Support Auto ID
- + Power—6 AA batteries/rechargeable



Brick Status Light – Red



Brick Status Light – Orange



Brick Status Light – Green

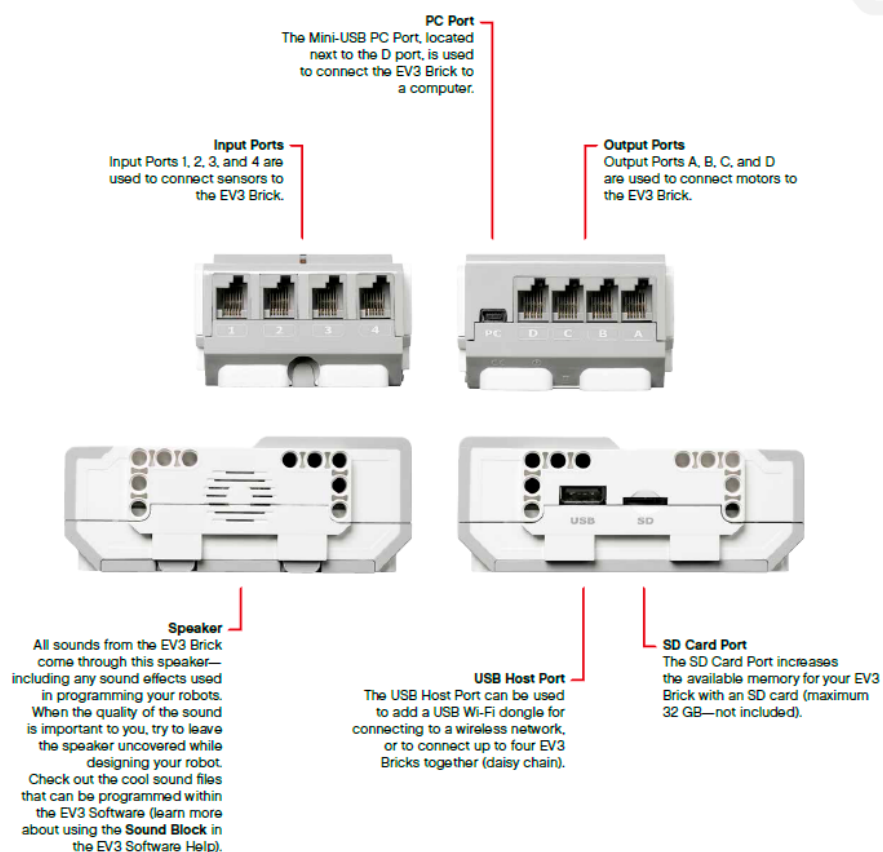
LEGOeducation.com

LEGO, the LEGO logo, MINDSTORMS and the MINDSTORMS logo are trademarks of the/ sont des marques de commerce de/son marcas registradas de LEGO Group. ©2013 The LEGO Group. 041329.

6

EV3 TECHNOLOGY

EV3 Brick



EV3 TECHNOLOGY

EV3 Motors

Large Motor

The Large Motor is a powerful "smart" motor. It has a built-in Rotation Sensor with 1-degree resolution for precise control. The Large Motor is optimized to be the driving base on your robots.

By using the Move Steering or Move Tank programming block in the EV3 Software, the Large Motors will coordinate the action simultaneously.

Medium Motor

The Medium Motor also includes a built-in Rotation Sensor (with 1-degree resolution), but it is smaller and lighter than the Large Motor. That means it is able to respond more quickly than the Large Motor.

The Medium Motor can be programmed to turn on or off, control its power level, or to run for a specified amount of time or rotations.

COMPARE THE TWO MOTORS:

- ✦ The Large Motor runs at 160–170 rpm, with a running torque of 20 Ncm and a stall torque of 40 Ncm (slower, but stronger).
- ✦ The Medium Motor runs at 240–250 rpm, with a running torque of 8 Ncm and a stall torque of 12 Ncm (faster, but less powerful).
- ✦ Both motors are Auto ID supported.

For more information on using the Rotation Sensor in your programming, see **Using the Motor Rotation Sensor** in the EV3 Software Help.



Large Motor



Medium Motor

EV3 TECHNOLOGY

EV3 Sensors

Color Sensor

The Color Sensor is a digital sensor that can detect the color or intensity of light that enters the small window on the face of the sensor. This sensor can be used in three different modes: Color Mode, Reflected Light Intensity Mode, and Ambient Light Intensity Mode.

In **Color Mode**, the Color Sensor recognizes seven colors—black, blue, green, yellow, red, white, and brown—plus No Color. This ability to differentiate between colors means your robot might be programmed to sort colored balls or blocks, speak the names of colors as they are detected, or stop action when it sees red.

In **Reflected Light Intensity Mode**, the Color Sensor measures the intensity of light reflected back from a red light-emitting lamp. The sensor uses a scale of 0 (very dark) to 100 (very light). This means your robot might be programmed to move around on a white surface until a black line is detected, or to interpret a color-coded identification card.

In **Ambient Light Intensity Mode**, the Color Sensor measures the strength of light that enters the window from its environment, such as sunlight or the beam of a flashlight. The sensor uses a scale of 0 (very dark) to 100 (very light). This means your robot might be programmed to set off an alarm when the sun rises in the morning, or stop action if the lights go out.

The sample rate of the Color Sensor is 1 kHz/sec.

For the best accuracy, when in Color Mode or Reflected Light Intensity Mode, the sensor must be held at a right angle, close to—but not touching—the surface it is examining.

For more information, see **Using the Color Sensor** in the EV3 Software Help.



Color Sensor



Color Mode



Reflected Light Intensity Mode



Ambient Light Intensity Mode

EV3 TECHNOLOGY

EV3 Sensors

Gyro Sensor

The Gyro Sensor is a digital sensor that detects rotational motion on a single axis. If you rotate the Gyro Sensor in the direction of the arrows on the case of the sensor, the sensor can detect the rate of rotation in degrees per second. (The sensor can measure a maximum rate of spin of 440 degrees per second.) You can then use the rotation rate to detect, for example, when a part of your robot is turning, or when your robot is falling over.

In addition, the Gyro Sensor keeps track of the total rotation angle in degrees. You can use this rotation angle to detect, for example, how far your robot has turned. This feature means you are able to program turns (on the axis the Gyro Sensor is measuring) with an accuracy of ± 3 degrees for a 90-degree turn.

Note: The sensor must be completely motionless while being plugged into the EV3 Brick. If the Gyro Sensor is attached to a robot, the robot should be held motionless in its starting position as the Gyro Sensor is plugged into the EV3 Brick.

CONNECTING THE GYRO SENSOR

On the EV3 Brick, go to the Brick App screen (third tab) and use the Center button to select Port View.

Using a flat black Connector Cable, connect the Gyro Sensor to the EV3 Brick on Port 2. Make sure to hold the sensor very still while you do this. On the EV3 Brick display, the Port View app should show a "0" reading in the second small bottom window from the left, which is the window representing input values from Port 2.

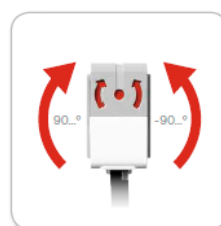
Still without moving the sensor, observe the display for a few seconds. It should continue to display "0" for Port 2 with the Gyro Sensor. In case the Gyro Sensor readings do not constantly display "0" during the connection process, unplug the sensor and repeat the procedure.

When the screen consistently displays "0" for a few seconds, experiment with rotating the sensor and observe how the angle reading changes. Remember, the Gyro Sensor measures the angle of change on only one axis.

For more information, see **Using the Gyro Sensor** in the EV3 Software Help.



Gyro Sensor



Single axis rotation



Brick App screen



Port View with Gyro Sensor

EV3 TECHNOLOGY

EV3 Sensors

Touch Sensor

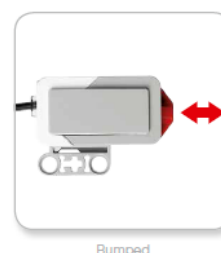
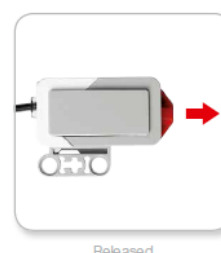
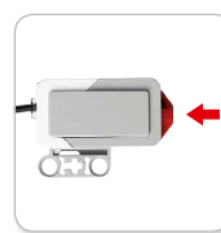
The Touch Sensor is an analog sensor that can detect when the sensor's red button has been pressed and when it is released. That means the Touch Sensor can be programmed to action using three conditions—pressed, released, or bumped (both pressed and released).

Using input from the Touch Sensor, a robot can be programmed to see the world as a blind person might, reaching a hand out and responding when it touches something (pressed).

You might build a robot with a Touch Sensor pressed against the surface beneath it. You might then program the robot to respond (Stop!) when your robot is about to drive off the edge of a table (when the sensor is released).

A fighting robot might be programmed to keep pushing forward against its challenger until the challenger retreats. That pair of actions—pressed, then released—would constitute bumped.

For more information, see **Using the Touch Sensor** in the EV3 Software Help.



EV3 TECHNOLOGY

EV3 Sensors

Infrared Sensor and Remote Infrared Beacon

The Infrared Sensor is a digital sensor that can detect infrared light reflected from solid objects. It can also detect infrared light signals sent from the Remote Infrared Beacon.

The Infrared Sensor can be used in three different modes: Proximity Mode, Beacon Mode, and Remote Mode.

PROXIMITY MODE

In Proximity Mode, the Infrared Sensor uses the light waves reflected back from an object to estimate the distance between the sensor and that object. It reports the distance using values between 0 (very close) to 100 (far away), not as a specific number of centimeters or inches. The sensor can detect objects up to 70 cm away, depending on the size and shape of the object.

For more information, see **Using the Infrared Sensor Proximity Mode** in the EV3 Help files.

BEACON MODE

Choose one of the Remote Infrared Beacon's four channels from the red Channel Selector. The Infrared Sensor will detect a beacon signal matching the channel that you specify in your program up to a distance of approximately 200 cm in the direction it is facing.

Once detected, the sensor can estimate the general direction (heading) of and the distance (proximity) to the beacon. With this information, you can program a robot to play hide-and-seek, using the Remote Infrared Beacon as the target of its search. The heading will be a value of between -25 and 25, with 0 indicating that the beacon is directly in front of the Infrared Sensor. The proximity will be values between 0 and 100.

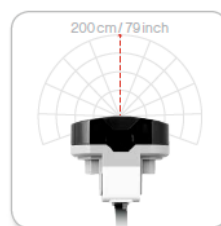
For more information, see **Using the Infrared Sensor Beacon Mode** in the EV3 Help files.



Infrared Sensor



Proximity Mode



Beacon Mode

EV3 TECHNOLOGY

EV3 Sensors

The Remote Infrared Beacon is a separate device that can be handheld or built into another LEGO® model. It requires two AAA alkaline batteries. To turn the Remote Infrared Beacon on, press the big Beacon Mode button on the top of the device. A green LED Indicator will turn on, indicating the device is active and transmitting continuously. Another press on the Beacon Mode button will turn it off (after one hour of inactivity, the beacon will automatically turn itself off).

REMOTE MODE

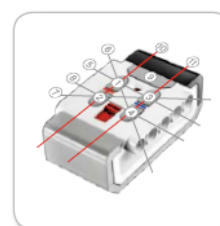
You can also use the Remote Infrared Beacon as a remote control for your robot. When in Remote Mode, the Infrared Sensor can detect which button (or combination of buttons) on the beacon is pressed. There are a total of eleven possible button combinations:

- 0 = No button (and Beacon Mode is off)
- 1 = Button 1
- 2 = Button 2
- 3 = Button 3
- 4 = Button 4
- 5 = Both Button 1 and Button 3
- 6 = Both Button 1 and Button 4
- 7 = Both Button 2 and Button 3
- 8 = Both Button 2 and Button 4
- 9 = Beacon Mode is on
- 10 = Both Button 1 and Button 2
- 11 = Both Button 3 and Button 4

For more information, see **Using the Infrared Sensor Remote Mode** in the EV3 Help files.



Remote Infrared Beacon



Remote Mode

NOTE:

The Infrared Sensor and Remote Infrared Beacon are not part of the LEGO® MINDSTORMS® Education EV3 Core Set but can be acquired as accessories.

Annex C. Codi Matlab

Per a que funcioni el codi, és necessari descarregar el fitxer *rscale.m* que es troba a [8]. S'ha utilitzat com a referent [6].

%DADES

```
g = 9.81;           %gravetat
mp = 0.552;         %massa pèndol
mc = 0.08;          %massa carro
l = 0.162;          %longitud del pèndol
mu = 0.00769;       %constant de fregament de l'eix
r = 0.0475;         %radi de la roda

st = 0.03;          %temps de mostratge
```

%OBTENCIÓ de K, Ñ i L

%matrius de l'espai d'estats

```
A = [0      1      0      0;
      0      0     -mp*g/(1.33*(mc+mp)-mp)  mu/(1*(1.33*(mc+mp)-mp));
      0      0      0      1;
      0      0     (mc+mp)*g/(1*(1.33*(mc+mp)-mp))  -mu/(mp*1*1*(1.33-
      (mp./(mc+mp))))];
```

```
B = [      0;
      (1/(mc+mp)) + (mp/((mc+mp)*(1.33*(mc+mp)-mp)));
      0;
      (-1/(1*(1.33*(mc+mp)-mp)))];
```

```
C = [1 0 0 0;
      0 1 0 0;
      0 0 1 0;
      0 0 0 1];
```

```
D = [0;
      0;
      0;
      0];
```

```
states = {'x' 'x_dot' 'phi' 'phi_dot'};
inputs = {'u'};
outputs = {'x'; 'x_dot'; 'phi'; 'phi_dot'};
```

```
sys_ss =
ss(A,B,C,D, 'statename', states, 'inputname', inputs, 'outputname', outputs);
```



```

%controlabilitat del sistema

co = ctrb(sys_ss);
controllability = rank(co)

%definició matrius LQR

Q = C'*C;
Q(1,1) = 1;
Q(2,2) = 1;
Q(3,3) = 1;
Q(4,4) = 1;
R = 1

K = lqr(A,B,Q,R)

Ac = [(A-B*K)];
Bc = [B];
Cc = [C];
Dc = [D];

states = {'x' 'x_dot' 'phi' 'phi_dot'};
inputs = {'r'};
outputs = {'x'; 'x_dot'; 'phi'; 'phi_dot'};

sys_cl =
ss(Ac,Bc,Cc,Dc,'statename',states,'inputname',inputs,'outputname',outputs);

%resposta amb LQR

figure
t = 0:0.01:5;
rp = 0.2*ones(size(t));
[y,t,x0]=lsim(sys_cl,rp,t);
[AX,H1,H2] = plotyy(t,y(:,1),t,y(:,3),'plot');
set(get(AX(1),'Ylabel'),'String','cart position (m)')
set(get(AX(2),'Ylabel'),'String','pendulum rotation(radians)')
title('Step Response with LQR Control')

%determinació del pre-compensador

Cn = [1 0 0 0];
sys_ss = ss(A,B,Cn,0);
Nbar = rscale(sys_ss,K)

sys_cl =
ss(Ac,Bc*Nbar,Cc,Dc,'statename',states,'inputname',inputs,'outputname',ou
tputs);

```

```

%resposta amb LQR i pre-compensador

figure
[y,t,x1]=lsim(sys_cl,rp,t);
[AX,H1,H2] = plotyy(t,y(:,1),t,y(:,3),'plot');
set(get(AX(1),'Ylabel'),'String','cart position (m)')
set(get(AX(2),'Ylabel'),'String','pendulum rotation(radians)')
title('Step Response with Precompensation and LQR Control')

%càlcul dels pols d'Ac per obtenir els d'L

poles = eig(Ac);

%controlabilitat del sistema

co = ctrb(sys_ss);
controllability = rank(co)

%pols L convergeixen 9 vegades més ràpid que Ac

p = round(9*(max(real(poles(poles(:,1)<0,:)))));
P = [p p-1 p-2 p-3];
L = place(A',C',P)';

Ace = [(A-B*K) (B*K);
        zeros(size(A)) (A-L*C)];
Bce = [B*Nbar;
        zeros(size(B))];
Cce = [Cc zeros(size(Cc))];
Dce = [0;0;0;0];

states = {'x' 'x_dot' 'phi' 'phi_dot' 'e1' 'e2' 'e3' 'e4'};
inputs = {'r'};
outputs = {'x'; 'x_dot'; 'phi'; 'phi_dot'};

sys_est_cl =
ss(Ace,Bce,Cce,Dce,'statename',states,'inputname',inputs,'outputname',out
puts);

%sistema amb LQR, pre-compensador i observaor
figure
[y,t,x2]=lsim(sys_est_cl,rp,t);
[AX,H1,H2] = plotyy(t,y(:,1),t,y(:,3),'plot');
set(get(AX(1),'Ylabel'),'String','cart position (m)')
set(get(AX(2),'Ylabel'),'String','pendulum rotation(radians)')
title('Step Response with Observer-Based State-Feedback Control')

```